

Neu

Erweiterungen für Amica-Paint

- Neue Maustreiber
- Verbesserte Diashow-Routine
- 2 zusätzliche Zeichensätze





Faszination Grafik

Dreidimensionale Körper drehen sich ästhetisch auf der Mattscheibe. Zwei- oder dreimal auf den Feuerknopf gedrückt, und in Windeseile berechnet der Computer Linien, Kreise oder Ellipsen. Was steckt dahinter? Wie programmiert man so etwas?

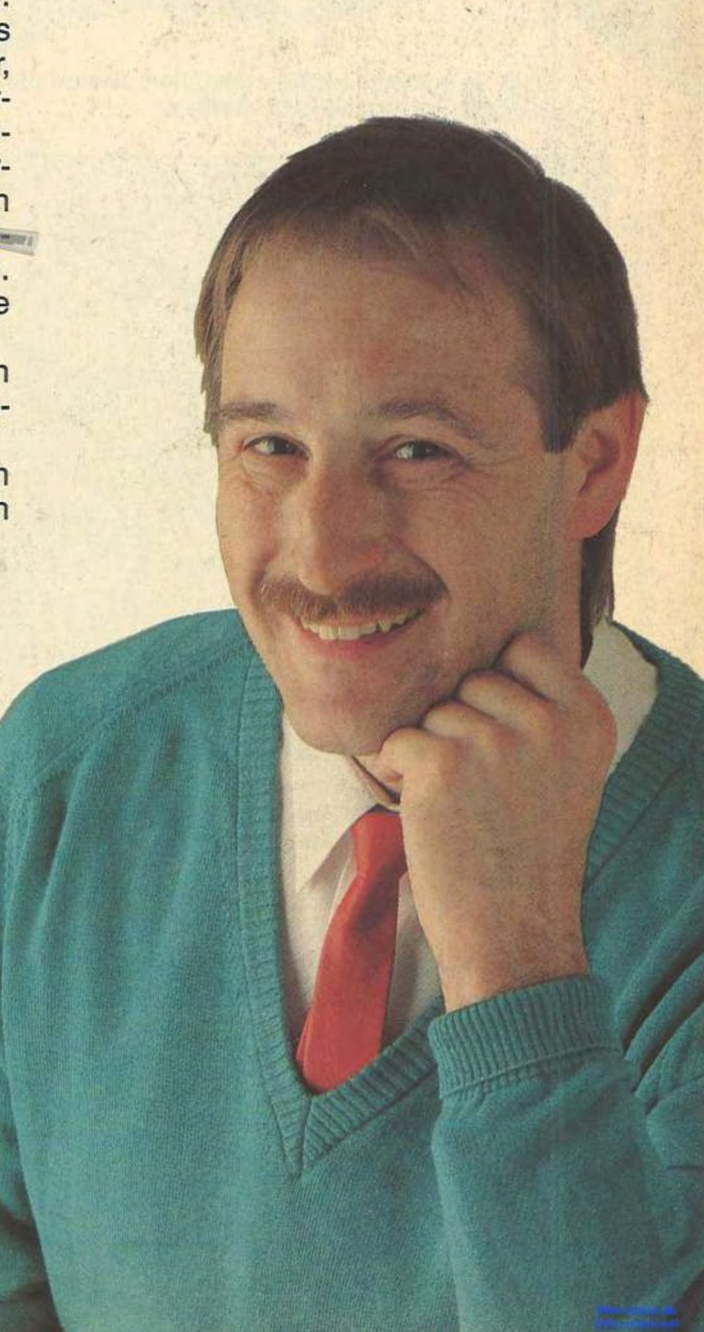
Wir haben den Profis über die Schulter geschaut und das Ganze in einem ausführlichen Grafikkurs zusammengefaßt. »Das verstehe ich ja nie!« Keine Angst, in unserem Kurs fangen wir ganz von vorn an – mit einfachen Programmen in Basic. Später erfahren Sie, wie man schnelle Grafikroutinen in Assembler programmiert bis hin zur Entwicklung von 3D-Effekten.

Von Workstations verwöhnte Profis winken ab: »Das ist doch kein Metier für den C 64 – dafür gibt es doch bessere, leistungsfähigere Computer«. Sicher, kein »Golf« wird über Nacht zu einem »Turbo-Porsche«. Aber wer kann sich letzteren schon leisten. Außerdem staunt man, was aus dem kleinen C 64 herauszuholen ist. Und die theoretischen Grundlagen der Grafikprogrammierung bleiben auch bei größeren Computern gleich oder zumindest sehr ähnlich. Wer einmal verstanden hat, wie gute Grafiksoftware entwickelt wird, profitiert immer wieder davon.

Also, keine Scheu vor dieser Materie, sondern »ran an die Buletten«. Probieren Sie selbst, welche Fähigkeiten in Ihrem Computer schlummern.

Lassen Sie sich begeistern von der faszinierenden Welt der Computergrafik. Ihre schönsten Grafiken oder pfiffige Programme können Sie ja an die Redaktion Sonderhefte schicken.

Ihr Gottfried Knechtel
(Stellv. Chefredakteur)

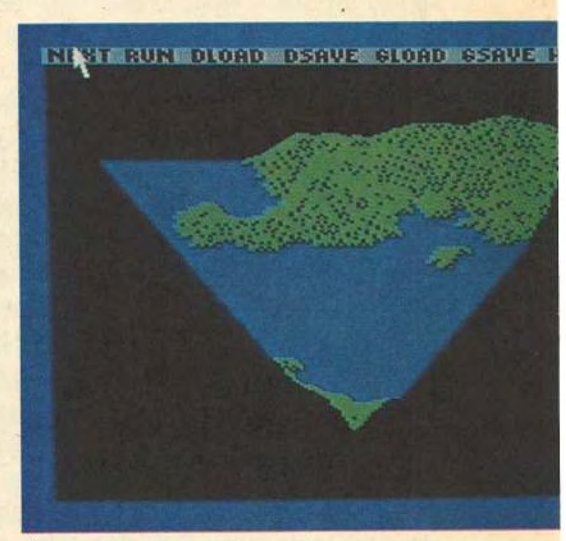


ALLES ÜBER GRAFIK

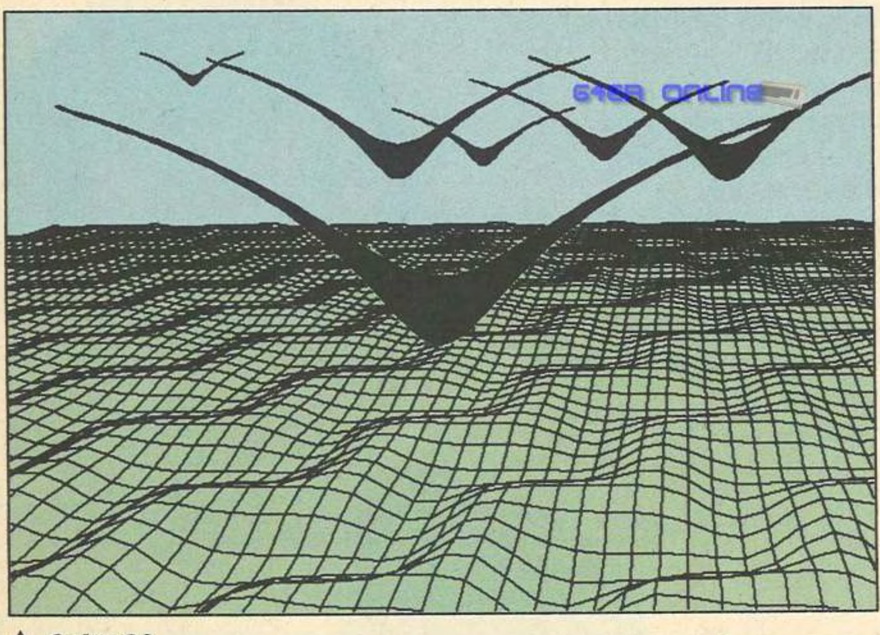
Sämtliche Grundlagen, die für die Grafikprogrammierung notwendig sind, wollen wir Ihnen hier vorstellen. Vor allem diejenigen, die neu in den Grafikbereich einsteigen, sind zum aktiven Mitmachen aufgerufen.

Alle Grafikmöglichkeiten des C64 basieren auf dem Video-Interface-Chip 6567 (auch bekannt als VIC-II-Chip). Dieser Chip ermöglicht die verschiedensten Grafikarten, einschließlich einer Textdarstellung von 40 Zeichen x 25 Zeilen, eine hochauflösende Grafik von 320 x 200 Punkten sowie Sprites, kleine bewegliche Objekte, die das Erstellen von Spielen wesentlich vereinfachen. Darüber hinaus können viele dieser Grafikarten auf demselben Bildschirm gemischt werden. Sprites lassen sich mit allen Displayarten kombinieren. Wir werden später noch auf Sprites genauer eingehen. Zunächst beschäftigen wir uns mit den übrigen Grafik... der mit dem VIC-II-Chip ... 97).

▲ In unserem umfassenden Kurs erfahren Sie alle Grundlagen zur Grafikprogrammierung. **Seite 6**



▲ **Seite 14**
Werden Sie zum »Landschaftsgärtner«: In wenigen Schritten entwickeln sich immer neue Berge und Seen



▲ **Seite 38**
Surrealistische Grafiken lassen sich mit »Prograf« aus mathematischen Funktionen entwickeln

Grafikkurs

Listings

Die Farbenpracht von »Apfelmännchen« versetzt jeden Betrachter immer wieder in Erstaunen.

Seite 8 ▶





Ein kleiner Umbau Ihres Plotters genügt, um fantastische Hardcopies zu erzeugen.

◀ Seite 60

JETZT WIRD´S



Tips & Tricks

Eingabehilfen

Sonstiges

Alle Programme aus Artikeln mit einem ■ -Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft

ALLES ÜBER GRAFIK

Einer der faszinierendsten Aspekte des C64 sind seine Grafikfähigkeiten. Mit dem Wissen, das Ihnen dieser Kurs vermittelt, werden auch Sie in der Lage sein, diese voll auszuschöpfen.

Sämtliche Grundlagen, die für die Grafikprogrammierung notwendig sind, wollen wir Ihnen hier vorstellen. Vor allem diejenigen, die neu in den Grafikbereich einsteigen, sind zum aktiven Mitmachen aufgerufen.

Alle Grafikmöglichkeiten des C64 basieren auf dem Video-Interface-Chip 6567 (auch bekannt als VIC-II-Chip). Dieser Chip ermöglicht die verschiedensten Grafikarten, einschließlich einer Textdarstellung von 40 Zeichen x 25 Zeilen, eine hochauflösende Grafik von 320 x 200 Punkten sowie Sprites, kleine bewegliche Objekte, die das Erstellen von Spielen wesentlich vereinfachen. Darüber hinaus können viele dieser Grafikarten auf demselben Bildschirm gemischt werden. Sprites lassen sich mit allen Displayarten kombinieren. Wir werden später noch auf Sprites genauer eingehen. Zunächst beschäftigen wir uns mit den übrigen Grafikarten. Eine Aufstellung der mit dem VIC-II-Chip möglichen Grafikarten zeigt Bild 1 (Seite 97).

Zunächst einige allgemeine Informationen. Der Bildschirm des C64 verfügt über 1000 Positionen. Normalerweise beginnt der Bildschirmspeicher bei Adresse 1024 ($0400 in hexadezimaler Darstellung) und reicht bis Adresse 2023 ($07E7). Jede dieser Adressen kann 8 Bit speichern, das entspricht einer beliebigen ganzen Zahl zwischen 0 und 255. Dem Bildschirmspeicher entspricht eine Gruppe von 1000 Adressen, die Farbspeicher oder Farb-RAM genannt wird. Dieser beginnt bei Adresse 55296

Schiebung im Speicher

($D800) und reicht bis Adresse 56295 ($DBE7). Die vier niederwertigen Bits dieser Adressen bestimmen die Farbe einer 8 x 8-Matrix im Bildschirm. Es sind also 2↑4 Bitkombinationen – was den Zahlen zwischen 0 und 15 entspricht –

Darüber hinaus können jederzeit 256 verschiedene Zeichen angezeigt werden. Bei der normalen Bildschirmanzeige enthält jede der 1000 Adressen des Bildschirmspeichers eine Code-Zahl, die dem VIC-II-Chip »sagt«, welches Zeichen an dieser Bildschirmposition anzuzeigen ist.

Die verschiedenen Grafikmodi werden über die 47 Steuerregister im VIC-II-Chip gewählt. Viele Grafikfunktionen lassen sich steuern, indem der richtige Wert über die POKE-Anweisung in eines der Register geschrieben wird. Die Register des VIC-II-Chip befinden sich an den Speichern 53248 ($D000 in Hexadezimaldarstellung) bis 53294 ($D02E).

Wahl der Video-Bank

Der VIC-II-Chip kann gleichzeitig auf einen Speicherbereich von 16 KByte zugreifen. Da der C64 über einen 64-KByte-Speicher verfügt, soll der VIC-II natürlich auch den ganzen Speicher »sehen« können. Dies ist möglich. Es gibt vier verschiedene Banks (oder Abschnitte), mit jeweils 16 KByte. Nun muß lediglich noch geregelt werden, auf welche dieser Abschnitte der VIC-II-Chip zugreift. Auf diese Weise kann der Chip die gesamte Speicherkapazität von 64 KByte »überblicken«. Die Bankanwahl-Bits, die einen Zugriff auf die verschiedenen Speicherabschnitte ermöglichen, befinden sich im Complex-Interface Adapter-Chip #2 (CIA #2) 6526. Über die Basic-Anweisungen POKE und PEEK (oder die entsprechenden Versionen in Maschi-

Fortsetzung auf Seite 97.

Kursübersicht

möglich. Jede der einzelnen Zahlen entspricht einer Farbe, und somit sind mit dem C64 16 verschiedene Farben darstellbar.

Apfelmännchen aus

Apfelmännchen-Grafiken versetzen den Betrachter immer wieder in Erstaunen. Dieses Erstaunen verwandelt sich in Begeisterung, wenn Sie die fantastischen Apfelmännchen in 16 Farben sehen, die Michael Aumers Programm erzeugt.



Viele »Apfelmännchen«-Programme bieten dem Anwender aufgrund mangelnder Farbgebung nicht die Qualität, die den Fähigkeiten des C64 entspricht. Erst durch die Vielfalt aller 16 darstellbaren Farben können die Strukturen der Mandelbrotmenge gut erkannt werden. Unser »Apfelmännchen in 16 Farben« nutzt die Grafikmöglichkeiten des C64 voll aus. Die Anwahl der einzelnen Programmabschnitte, wie Erzeugen und Darstellen der Grafiken, ist menügesteuert. Dadurch kommen auch alle, die sich noch nicht mit der Theorie der Mandelbrot-Mengen auskennen, in den vollen Genuß der herrlichen Bilder.

Bedienungshinweise

Das Programm bietet bei fast allen Funktionen Hinweise zu den Möglichkeiten, die zur Verfügung stehen. Um Ihnen die Bedienung so leicht wie möglich zu machen, erklären wir Ihnen jetzt ausführlich alle Menü-Punkte.

Nach dem Start des Basic-Programms »Apfel16.bas« (Listing 1) mit RUN erscheint das Hauptmenü mit der Überschrift »Apfelmännchen !«. Die einzelnen Menü-Punkte werden mit den Tasten 1 bis 7 aufgerufen. Folgende Auswahlmöglichkeiten sind vorhanden:

<1> **Daten eingeben/anzeigen:** Hier werden die aktuellen Werte zur Berechnung (Werte für die Ränder und die Iterationsgrenze) angezeigt. Beim Laden des Basic-Programms sind bereits Werte voreingestellt. Wollen Sie die angezeigten Werte übernehmen, ist einfach die Taste <RETURN> zu drücken.

<2> **Bild laden:** Sie können erstellte Bilder auf Diskette speichern und mit diesem Menüpunkt wieder laden. Im Inhaltsverzeichnis der Diskette erkennen Sie Bilder an dem Nachsatz ».pic«. Achtung: Dieser Nachsatz darf beim Aufrufen der Bilder nicht mit angegeben werden.

<3> **Bild speichern:** Mit dieser Auswahl speichern Sie ein Bild unter einem von Ihnen festgelegten Namen auf Diskette. Der Name darf maximal zwölf Zeichen lang sein. Ein Bild belegt auf Diskette 41 Blocks, diese müssen vor dem Speichern also frei sein.

dem Farbtopf gezogen

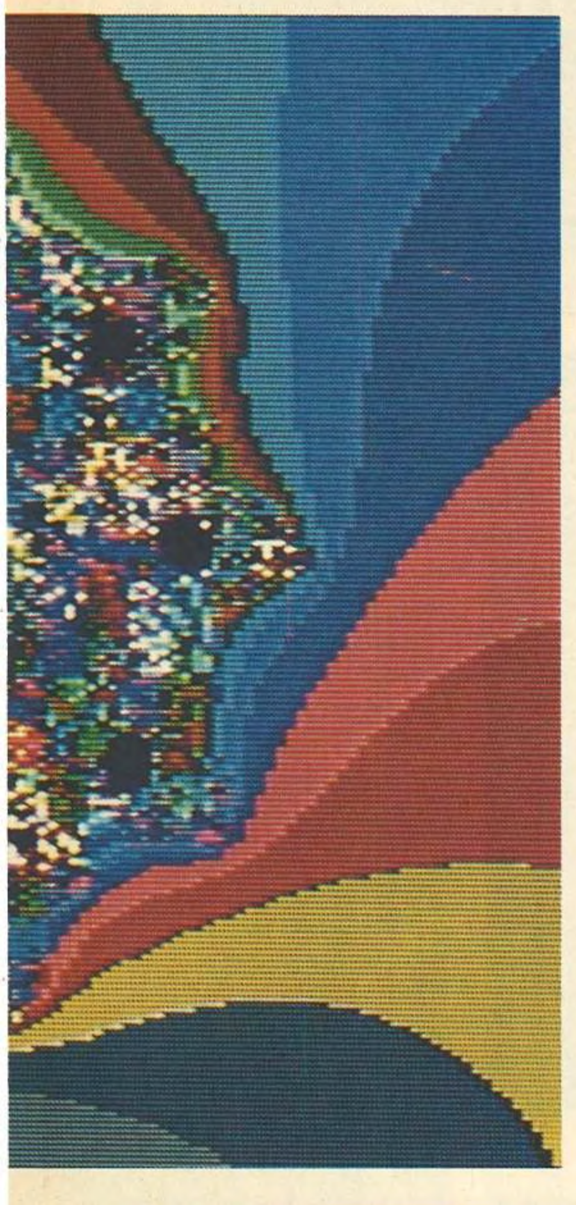


Bild 1. Ein »Apfelmännchen« mit 16 Farben

<4> Bild berechnen: Weist das Programm an, ein Bild mit den Daten, die Sie unter Menüpunkt <1> eingegeben haben, neu zu berechnen. Die Berechnungszeit nimmt in Abhängigkeit von der Größe der eingegebenen Iterationsgrenze zu. Verwenden Sie die beim Programmstart vorhandenen Werte, können Sie eine Stunde am Bildschirm die Entstehung der Grafik bewundern oder eine Tasse Kaffee trinken – Ihr Computer ist jedenfalls beschäftigt.

<5> **Ausschnitt berechnen:** Ein beliebiger Ausschnitt der Grafik wird neu berechnet, ein neues Bild entsteht. Sie legen den Ausschnitt fest, indem Sie das am linken Bildrand erscheinende Rechteck mit den Tasten <CRSR-rechts> und <CRSR-links> bewegen. Die Taste <←> verändert die Geschwindigkeit der Bewegung. Mit den Tasten <+> und <-> verändern Sie die Größe des dargestellten Ausschnitts. <↑> legt den Ausschnitt fest. Nach Eingabe einer Iterationsgrenze beginnt die Berechnung der neuen Grafik.

<6> **Bild anzeigen/Farbeffekte:** Mit der Anwahl dieser Funktion gelangen Sie ins Untermenü »Farbauswahl«.

<7> **Directory:** Zeigt das Inhaltsverzeichnis der eingelegten Diskette an. Gespeicherte Bilder sind an einem angehängten ».pic« zu erkennen. Diese Namenserweiterung darf beim Laden der Bilder unter Punkt <2> keinesfalls mit angegeben werden.

Menü 2 – Farbkomposition

Im Untermenü »Farbauswahl«, das durch Drücken der Taste <6> aus dem Hauptmenü aufgerufen wird, legt man die Farbwerte für das angezeigte Bild fest. Die Grafiken können in 4 oder in 16 Farben dargestellt werden. Wählen Sie mit <3> die 16farbige Darstellung, können mit der Taste <R> die Farben zyklisch vertauscht werden. Mit <↑> kommen Sie zurück in das Untermenü.

Mit <4> wird die Darstellung in vier Farben ausgewählt. Die Farben können in diesem Modus mit den Funktionstasten verändert werden. Mit der Taste <↑> wird dieser Modus wieder verlassen.

Noch ein Tip für die 16farbige Darstellung: Schalten Sie doch einmal die Tastenwiederholung ein, indem Sie das Basic-Programm »Apfelm 16.bas« wie folgt ändern:

```
1425 poke 650,128
1430 poke 650,0
```

Die Farben wechseln dadurch so lange, wie Sie die Taste <R> gedückt halten.

Die Farbeffekte können Sie weiter durch die Menüpunkte »Farben sichern«, »Farben zurückholen« und »Farbgebung ändern« beeinflussen.

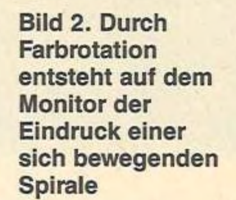


Bild 2. Durch Farbrotation entsteht auf dem Monitor der Eindruck einer sich bewegenden Spirale

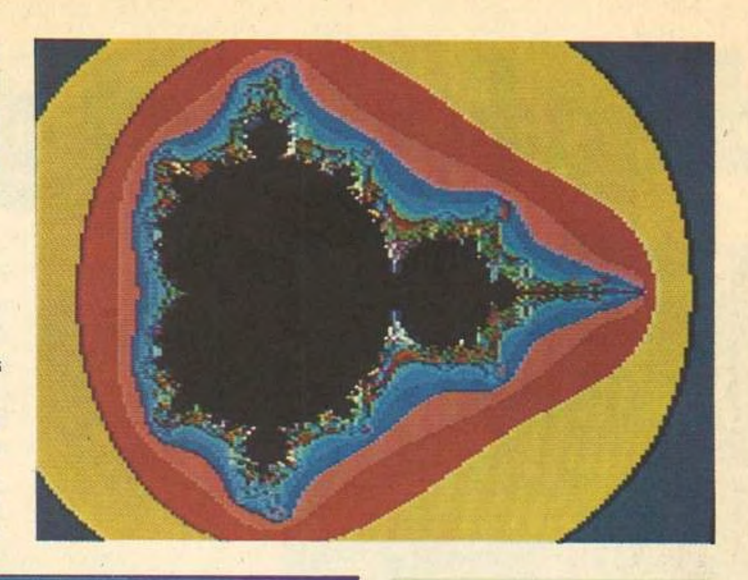


Kurzinfo: Apfelmännchen

Programmart: Tool zum Generieren von »Apfelmännchen«
Laden: LOAD "APFPEL16.BAS",8
Start: Nach dem Laden RUN eingeben
Besonderheiten: Das Basic-Programm lädt das Maschinensprache-Programm »MS.APFEL16« nach. Die Menüauswahl erfolgt über die Tastatur. Fertige Bilder befinden sich auf der Programmservice-Diskette.
Programmautor: Michael Aumer

Die Farbwerte, die nach dem Erstellen oder dem Laden eines Bildes vorliegen, werden mit Punkt <1>, »Farben sichern«, im Speicher abgelegt. Sollten Ihnen die Farbänderungen, die Sie mit anderen Funktionen durchführen, nicht gefallen, holen Sie mit der Funktion <2> die gespeicherte Farbkombination wieder zurück. Das Aufrufen des Menüpunktes <5> »Farbgebung ändern« gestattet Ihnen, die gesamte Farbzusammenstellung für die Grafik zu verändern. Hier wird auch die Hintergrundfarbe neu bestimmt. Die eingegebenen Werte müssen zwischen 0 und 15 liegen. Jede dieser Zahlen entspricht einer zugehörigen Farbe (Tabelle 1). Weitere Veränderungen der Farbumgebung entstehen, wenn Sie die gespeicherten Farben zurückholen und dann die Taste <R> drücken. Dadurch ändert sich die Farbfolge der Rotation im 16-Farben-Modus.

Funktionsweise des Programms



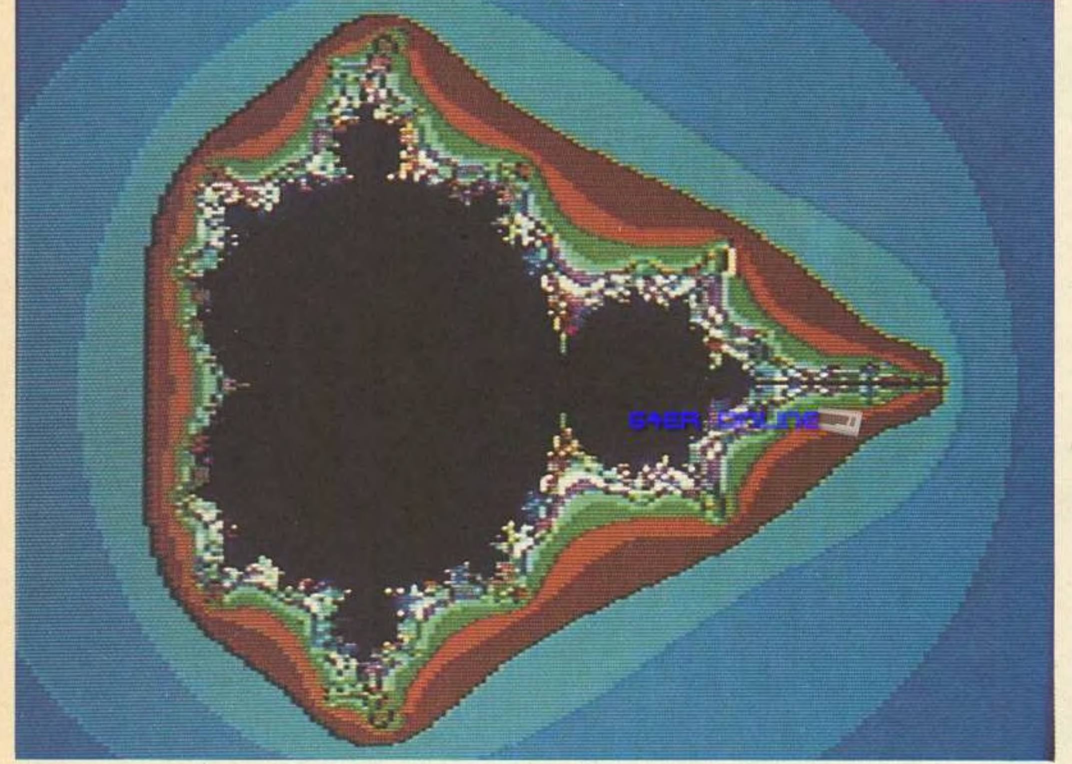


Bild 3. Wie eine Körperzelle wirkt dieses »Apfelmännchen«

Bild 4. Die Farbänderungen erzeugen skurrile Effekte

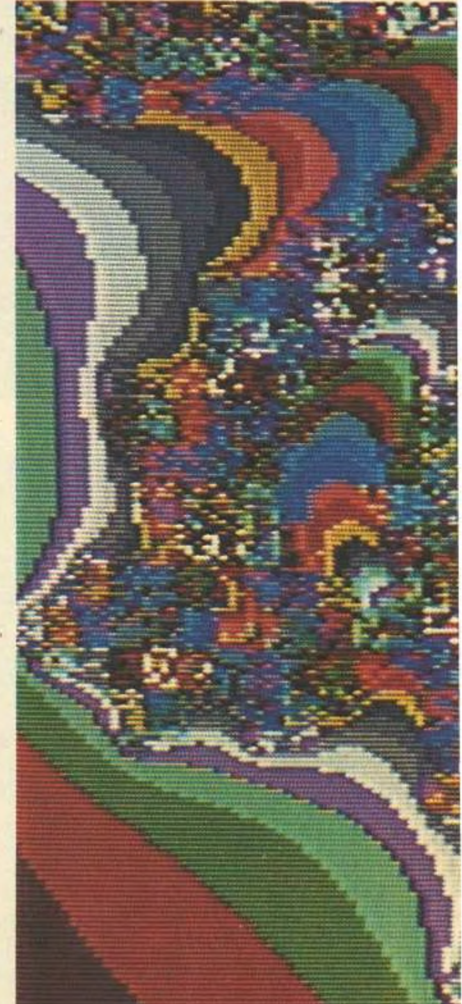


Bild 5. Durch den Farbverlauf entsteht der Eindruck räumlicher Tiefe

0	Schwarz	8	Orange
1	Weiß	9	Braun
2	Rot	10	Hellrot
3	Türkis	11	Grau 1
4	Violett	12	Grau 2
5	Grün	13	Hellgrün
6	Blau	14	Hellblau
7	Gelb	15	Grau 3

Tabelle 1. Hier finden Sie die Werte für die 16 Farben des C64

Das Programm stellt die Mandelbrotmenge (Apfelmännchen) dar. Dabei werden die Bildschirmkoordinaten in komplexe Zahlen umgewandelt und mit Hilfe einer Iterationsformel (z:= z hoch 2-c) ausgewertet. Die Berechnung des Wertes z erfolgt innerhalb einer Schleife. Die Abbruchbedingung für diese Berechnung ist das Erreichen einer oberen Grenze (8) für den Wert z. Wird dieser Wert überschritten, verläßt das Programm die Schleife. Einige Startwerte erreichen jedoch diesen Grenzwert erst sehr spät oder nie. Damit das Programm nicht ewig diese Schleife durchläuft, muß eine Iterationsgrenze angegeben werden. Diese bestimmt, wieviel Schleifendurchläufe maximal erfolgen. In dem vorliegenden Programm darf der Wert für die Iterationsgrenze 255 nicht überschreiten. Erreicht die Zahl der Schleifendurchläufe die eingegebene Iterationsgrenze,

Farbenpracht auf dem Bildschirm

bricht das Programmm ab und geht über zur nächsten Berechnung.

Damit Sie auch in den Genuß der 16farbigen Apfelmännchen-Bilder gelangen, geben Sie bitte das Listing 1 mit dem Checksummer ein. Dieses Basic-Programm ruft nach dem Starten das Maschinenprogramm »MS.Apfel16« (Listing 2) auf. Dieses muß sich unbedingt auf der Diskette im Laufwerk befinden und ist mit dem MSE einzugeben. Fertige Bilder finden Sie auf der Programmservice-Diskette. Wir wollen Ihnen nicht zumuten, 41 Blocks abzutippen, die fast nur aus den Folgen »00«, »FF« oder »55« bestehen.

(Michael Aumer/ef)



```
10 PRINT"{CLR,YELLOW}":POKE 53280,11:POKE
   53281,0:POKE 55,0:POKE 56,64                  <086>
20 IF LA=0 THEN LA=1:GOSUB 130:LOAD"MS.APF
   EL16",8,1                                     <209>
30 OPEN 5,0:POKE 786,192:V=53248:POKE V+39
   ,1:POKE V+40,1                                <142>
40 XU=-1:XO=2.4:YU=-1.25:YO=1.25:IT=40:GOS
   UB 60:GOTO 300                                <030>
50 REM ----- WRITE ------                        <008>
60 POKE 785,15:POKE 50683,80:XU=USR(XU):PO
   KE 50683,85:XO=USR(XO)                        <211>
70 POKE 50683,90:YU=USR(YU):POKE 50683,95:
   YO=USR(YO)                                    <005>
80 POKE 24420,IT:RETURN                          <116>
90 REM ---- LESEN ------                         <058>
100 POKE 785,12:POKE 50676,80:XU=USR(XU):P
    OKE 50676,85:XO=USR(XO)                      <067>
110 POKE 50676,90:YU=USR(YU):POKE 50676,95
    :YO=USR(YO)                                  <120>
120 IT=PEEK(24420):RETURN                        <070>
130 FOR N=28672 TO 28734:READ A:POKE N,A:N
    EXT                                          <236>
140 FOR N=28736 TO 28798:READ A:POKE N,A:N
    EXT:RETURN                                   <090>
150 DATA 192,0,0,192,0,0,192,0,0                 <055>
160 DATA 192,0,0,192,0,0,192,0,0                 <065>
170 DATA 192,0,0,192,0,0,192,0,0                 <075>
180 DATA 192,0,0,192,0,0,192,0,0                 <085>
190 DATA 192,0,0,192,0,0,192,0,0                 <095>
200 DATA 192,0,0,192,0,0,192,0,0                 <105>
210 DATA 192,0,0,192,0,0,255,255,255             <233>
220 REM -----------------                        <123>
230 DATA 255,255,255,0,0,3,0,0,3                 <240>
240 DATA 0,0,3,0,0,3,0,0,3                       <200>
250 DATA 0,0,3,0,0,3,0,0,3                       <210>
260 DATA 0,0,3,0,0,3,0,0,3                       <222>
270 DATA 0,0;3,0,0,3,0,0,3                       <232>
280 DATA 0,0,3,0,0,3,0,0,3                       <242>
290 DATA 0,0,3,0,0,3,0,0,3                       <252>
300 PRINT"{CLR,YELLOW,CTRL-H,CTRL-N}":POKE
     53280,11:POKE 53281,0                       <232>
310 PRINT"{5SPACE}APFELMAENNCHEN!{4SPACE,P
    URPLE}VON M. AUMER{YELLOW,2DOWN}"            <096>
320 PRINT" DATEN EINGEBEN / ANZEIGEN .....
     < 1 >{DOWN}"                                <098>
330 PRINT" BILD LADEN ....................
     < 2 >{DOWN}"                                <077>
340 PRINT" BILD SPEICHERN ................
     < 3 >{DOWN}"                                <048>
350 PRINT" BILD BERECHNEN ................
     < 4 >{DOWN}"                                <230>
360 PRINT" AUSSCHNITT BERECHNEN ..........
     < 5 >{DOWN}"                                <152>
370 PRINT" BILD ANZEIGEN / FARBEFFEKTE ...
     < 6 >{DOWN}"                                <089>
380 PRINT" DIRECTORY .....................
     < 7 >{DOWN}"                                <014>
390 POKE 198,0:WAIT 198,1:GET A$:POKE 198,
    0:Z=VAL(A$):IF Z=0 OR Z>7 THEN 390           <062>
400 ON Z GOSUB 450,610,680,750,770,1210,42
    0                                            <049>
410 GOTO 300                                     <100>
420 PRINT"{CLR}";:SYS 49161:POKE 198,0:WAI
    T 198,1:POKE 198,0:RETURN                    <172>
430 REM AENDERN                                  <182>
440 REM AENDERN                                  <192>
450 PRINT"{CLR}"                                 <184>
460 PRINT" NEUE EINGABEN VERAENDERN DIE WE
    RTE."                                        <169>
470 PRINT" NUR RETURN VERAENDERT NICHTS.{2
    DOWN}"                                       <231>
480 GOSUB 100                                    <170>
490 PRINT" LINKER{2SPACE}RAND:";XU               <017>
500 PRINT"{UP,14RIGHT}";:INPUT#5,XU:PRINT        <038>
510 PRINT" RECHTER RAND:";XO                     <115>
520 PRINT"{UP,14RIGHT}";:INPUT#5,XO:PRINT        <036>
530 PRINT" UNTERER RAND:";YU                     <207>
540 PRINT"{UP,14RIGHT}";:INPUT#5,YU:PRINT        <082>
550 PRINT" OBERER{2SPACE}RAND:";YO               <202>
560 PRINT"{UP,14RIGHT}";:INPUT#5,YO:PRINT:
    PRINT                                        <251>
```

Listing 1.
Dieses Basic-Programm steuert die Funktionen des »Apfelmännchen«-Programms

```
570 PRINT" ITERATIONSGRENZE <255! :";IT     <157>
580 PRINT"{UP,25RIGHT}";:INPUT#5,IT:PRINT:
    IF IT>255 OR IT<1 THEN 580              <180>
590 GOSUB 60:RETURN                         <038>
600 REM LADEN                               <016>
610 PRINT"{CLR,DOWN,RIGHT}LADEN{2DOWN}":PR
    INT" BITTE NAME OHNE .PIC EINGEBEN."    <102>
620 PRINT" MAXIMAL 12 ZEICHEN.{DOWN}":NA$=
    ""                                      <255>
630 PRINT" RETURN = ENDE !{3DOWN}"          <106>
640 INPUT#5,NA$:PRINT:PRINT:IF LEN(NA$)>12
    THEN RETURN                             <060>
650 IF NA$=""THEN RETURN                    <046>
660 A$=NA$+".PIC":SYS 49155,A$,8,1:RETURN   <174>
670 REM SPEICHERN                           <167>
680 PRINT"{CLR,DOWN,RIGHT}SPEICHERN{2DOWN}
    ":PRINT" BITTE NAME OHNE .PIC EINGEBEN
    ."                                      <117>
690 PRINT" MAXIMAL 12 ZEICHEN.{DOWN}":NA$=
    ""                                      <069>
700 PRINT" RETURN = ENDE !{3DOWN}"          <176>
710 INPUT#5,NA$:PRINT:PRINT:IF LEN(NA$)>12
    THEN RETURN                             <130>
720 IF NA$=""THEN RETURN                    <116>
730 A$=NA$+".PIC":SYS 49158,A$,8,1:RETURN   <085>
740 REM BERECHNEN                           <051>
750 PRINT"{CLR}";:POKE 53280,0:POKE 53281,
    0:POKE 24421,0:POKE 24422,0:SYS 49173:
    SYS 49176                               <044>
760 SYS 49170:SYS 49185:SYS 49152:SYS 4919
    1:SYS 49188:RETURN                      <048>
770 PRINT"{CLR}"                            <252>
780 PRINT" STEUERUNG DES KASTENS MITTELS C
    URSOR-"                                 <213>
790 PRINT" TASTEN.{DOWN}"                   <065>
800 PRINT" + UND - VERGROESSERT / VERKLEIN
    ERT DEN"                                <233>
810 PRINT"{9SPACE}KASTEN.{DOWN}"            <204>
820 PRINT" ← BESCHLEUNIGT ALLE BEWEGUNGEN   <131>
830 PRINT"{3SPACE}NOCHMALIGES DRUECKEN HEB
    T DIE                                   <206>
840 PRINT"{3SPACE}BESCHLEUNIGUNG WIEDER AU
    F !{DOWN}"                              <112>
850 PRINT" ↑ BEENDET DEN VORGANG."          <232>
860 GOSUB 100:DX=(XO-XU)/320:DY=(YO-YU)/20
    0                                       <011>
870 XX=0:YY=0:KY=24:KX=KY*1.6               <041>
880 POKE 198,0:WAIT 198,1:POKE 198,0        <221>
890 DD=1:SYS 49185:POKE V+21,3:POKE 25592,
    192:POKE 25593,193                      <105>
900 KX=KY*1.6:XS=XX+24:YS=YY+KY+30:X2=XX+K
    X+1:Y2=YY+50                            <066>
910 S=-2*(X2>255)-(XS>255)                  <060>
920 POKE V+1,YS:POKE V+3,Y2:POKE V,XS AND
    255:POKE V+2,X2 AND 255:POKE V+16,S     <206>
930 GET A$:IF A$=""THEN 930                 <234>
940 IF A$="←"THEN DD=9-DD                   <255>
950 IF A$="↑"THEN 1030                      <050>
960 IF A$="{DOWN}"THEN 1080                 <229>
970 IF A$="{UP}"THEN 1100                   <183>
980 IF A$="{LEFT}"THEN 1120                 <231>
990 IF A$="{RIGHT}"THEN 1140                <209>
1000 IF A$="+"THEN 1160                     <002>
1010 IF A$="-"THEN 1190                     <061>
1020 GOTO 930                               <042>
1030 XU=XU+DX*XX:YU=YU+DY*YY:XO=XU+DX*KX:Y
     O=YU+DY*KY:POKE V+21,0:SYS 49188:GOSU
     B 60                                   <158>
1040 PRINT"{CLR,CTRL-N}":PRINT" ITERATIONS
     GRENZE:";IT                            <170>
1050 PRINT"{UP,18RIGHT}";:INPUT#5,IT:PRINT  <197>
1060 IF IT>255 OR IT<1 THEN 1050            <222>
1070 POKE 24420,IT:GOTO 750                 <085>
1080 IF YS+DD>255 THEN 930                  <012>
1090 YY=YY+DD:GOTO 900                      <031>
1100 IF Y2-DD<0 THEN 930                    <202>
1110 YY=YY-DD:GOTO 900                      <179>
1120 IF XS-DD<0 THEN 930                    <232>
1130 XX=XX-DD:GOTO 900                      <091>
1140 IF X2+DD>511 THEN 930                  <004>
1150 XX=XX+DD:GOTO 900                      <239>
1160 IF X2+DD*1.6>511 THEN 930              <173>
1170 IF Y1+DD>255 THEN 930                  <068>
1180 KY=KY+DD:GOTO 900                      <127>
1190 IF KY-DD<2 THEN 930                    <042>
1200 KY=KY-DD:GOTO 900                      <019>
1210 PRINT"{CLR,YELLOW,CTRL-N}":POKE 53280
     ,14:POKE 53281,0                       <226>
1220 PRINT"{8SPACE}ANZEIGEMENUE !{2DOWN}"   <098>
1230 PRINT" FARBEN SICHERN ..............
     .. < 1 >{DOWN}"                        <109>
1240 PRINT" FARBEN ZURUECK HOLEN .........
     .. < 2 >{DOWN}"                        <062>
1250 PRINT" BILD IN 16 FARBEN ZEIGEN .....
     .. < 3 >{DOWN}"                        <253>
1260 PRINT" BILD IN 4{2SPACE}FARBEN ZEIGEN
     ...... < 4 >{DOWN}"                    <002>
1270 PRINT" FARBGEBUNG AENDERN ..........
     .. < 5 >{DOWN}"                        <116>
1280 PRINT" MENUE2 VERLASSEN ............
     .. < 6 >{DOWN}"                        <153>
1290 POKE 198,0:WAIT 198,1:GET A$:POKE 198
     ,0                                     <046>
1300 Z=VAL(A$):IF Z=0 OR Z>6 THEN 1290      <160>
1310 IF Z=6 THEN RETURN                     <073>
1320 ON Z GOSUB 1340,1350,1360,1470,1620    <071>
1330 GOTO 1210                              <064>
1340 SYS 49191:RETURN                       <087>
1350 SYS 49194:RETURN                       <227>
1360 PRINT"{CLR}"                           <078>
1370 PRINT" R ROTIERT EINMAL{SPACE,DOWN}"   <233>
1380 PRINT" ↑ ZURUECK IN MENUE2"            <011>
1390 POKE 198,0:WAIT 198,1:POKE 198,0       <223>
1400 POKE 53280,0:POKE 53281,RR:SYS 49185   <177>
1410 R=0                                    <109>
1420 REM                                    <212>
1430 GET A$:IF A$=""THEN 1430               <211>
1440 IF A$="R"THEN SYS 49182:GOTO 1430      <130>
1450 IF A$="↑"THEN SYS 49188:RETURN         <001>
1460 GOTO 1420                              <002>
1470 PRINT"{CLR}"                           <188>
1480 PRINT" F1 SCHALTET FARBE1 WEITER"      <146>
1490 PRINT" F3 SCHALTET FARBE2 WEITER"      <228>
1500 PRINT" F5 SCHALTET FARBE3 WEITER"      <054>
1510 PRINT" F7 SCHALTET HINTERGRUND WEITER
     "                                      <049>
1520 PRINT"{DOWN,SPACE}↑ = ZURUECK ZUM MEN
     UE2"                                   <039>
1530 A=106:B=7:C=0:POKE 198,0:WAIT 198,1:P
     OKE 198,0                              <058>
1540 SYS 49185:POKE 24421,A:POKE 24422,B:S
     YS 49173:POKE 53280,0:POKE 53281,C     <172>
1550 GET A$:IF A$=""THEN 1550               <142>
1560 IF A$="↑"THEN SYS 49188:SYS 49194:RET
     URN                                    <028>
1570 IF A$="{F7}"THEN C=(15 AND(C+1)):POKE
     53281,C:GOTO 1550                      <174>
1580 IF A$="{F5}"THEN B=(15 AND(B+1)):POKE
     24422,B:SYS 49173:GOTO 1550            <251>
1590 IF A$="{F1}"THEN A=(255 AND(A+16)):PO
     KE 24421,A:SYS 49173:GOTO 1550         <229>
1600 IF A$="{F3}"THEN A=(A AND 240)+(15 AN
     D(1+(A AND 15))):POKE 24421,A:SYS 491
     73:GOTO 1550                           <177>
1610 GOTO 1550                              <010>
1620 PRINT"{CLR}"                           <084>
1630 PRINT" BITTE GEBEN SIE DIE NEUEN FARB
     EN EIN !"                              <176>
1640 PRINT" HINTERGRUND:";:INPUT#5,RR:PRIN
     T                                      <231>
1650 FOR N=1 TO 15                          <061>
1660 PRINT" FARBE"N;TAB(17)":";:INPUT#5,A:
     PRINT                                  <076>
1670 POKE 50343+N,A:NEXT                    <155>
1680 SYS 49179:RETURN                       <048>
```


Listing 1.
Hauptprogramm »Apfelmännchen« (Schluß)

```
Name : ms.apfel16              c000 c601
-----------------------------------------
c000 : 4c 5f c1 4c 12 c4 4c 30   cf
c008 : c4 4c b1 c3 4c f3 c5 4c   eb
c010 : fa c5 4c 49 c4 4c 68 c4   03
c018 : 4c c7 c4 4c e4 c4 4c d3   50
c020 : c4 4c b9 c5 4c d6 c5 4c   5c
c028 : 91 c5 4c a5 c5 00 40 80   c2
c030 : c0 00 10 20 30 00 04 08   1c
c038 : 0c 00 01 02 03 00 00 00   f5
c040 : ff ff ff ff ff ff ff ff   3f
c048 : ff ff ff ff ff ff ff ff   47
c050 : ff ff ff ff ff ff ff ff   4f
c058 : ff ff ff ff ff ff ff 7f   56
c060 : 00 00 00 00 00 00 00 00   61
c068 : 00 00 00 00 00 ac 60 c0   d1
c070 : 20 a2 b3 a9 3d a0 c0 20   1f
c078 : 28 ba a9 50 a0 5f 20 67   c6
c080 : b8 20 b4 bf a2 47 a0 c0   d6
c088 : 20 d4 bb ac 61 c0 20 a2   79
c090 : b3 a9 42 a0 c0 20 28 ba   e0
c098 : a9 5a a0 5f 20 67 b8 20   e3
c0a0 : b4 bf a2 4c a0 c0 20 d4   a0
c0a8 : bb a2 0a a9 00 9d 51 c0   20
c0b0 : ca 10 fa a9 01 8d 5b c0   e2
c0b8 : 8d 5c c0 a9 ff 8d 5d c0   3c
c0c0 : a9 51 a0 c0 20 a2 bb a9   ab
c0c8 : 51 a0 c0 20 28 ba 20 ca   0c
c0d0 : bb a9 56 a0 c0 20 a2 bb   18
c0d8 : a9 56 a0 c0 20 28 ba 20   5b
c0e0 : c7 bb a9 57 a0 00 20 67   33
c0e8 : b8 a5 61 c9 84 90 02 18   09
c0f0 : 60 a9 51 a0 c0 20 a2 bb   9c
c0f8 : e6 61 a9 56 a0 c0 20 28   a5
c100 : ba a9 4c a0 c0 20 67 b8   d2
c108 : a2 56 a0 c0 20 d4 bb a9   00
c110 : 5c a0 00 20 a2 bb a9 57   1e
c118 : a0 00 20 50 b8 a9 47 a0   02
c120 : c0 20 67 b8 a2 51 a0 c0   9a
c128 : 20 d4 bb ee 5d c0 ad 5d   cc
c130 : c0 cd 64 5f 90 01 60 a0   b0
c138 : 01 ee 5c c0 ad 5c c0 c9   34
c140 : 04 90 03 8c 5c c0 ee 5b   1d
c148 : c0 ad 5b c0 c9 10 90 03   33
c150 : 8c 5b c0 4c c0 c0 a5 66   b9
c158 : 45 6e 85 6f a5 61 60 78   fb
c160 : a0 00 8c 6b c0 8c 6c c0   34
c168 : 8c 5f c0 8c 5e c0 a9 50   99
c170 : a0 5f 20 a2 bb a9 55 a0   bc
c178 : 5f 20 50 b8 20 0c bc a0   a9
c180 : a0 20 a2 b3 20 56 c1 20   4b
c188 : 12 bb a2 3d a0 c0 20 d4   02
c190 : bb a9 5a a0 5f 20 a2 bb   c3
c198 : a9 5f a0 5f 20 50 b8 20   ad
c1a0 : 0c bc a0 c8 20 a2 b3 20   72
c1a8 : 56 c1 20 12 bb a2 42 a0   44
c1b0 : c0 20 d4 bb 18 ad 5e c0   17
c1b8 : 8d 60 c0 ad 5f c0 8d 61   50
c1c0 : c0 20 6d c0 ad 5d c0 8d   e8
c1c8 : 62 c0 ad 5e c0 09 03 8d   3d
c1d0 : 60 c0 20 6d c0 ad 5d c0   b6
c1d8 : 8d 63 c0 ad 5f c0 09 07   2b
c1e0 : 8d 61 c0 20 6d c0 ad 5d   a0
c1e8 : c0 8d 64 c0 ad 5e c0 8d   8c
c1f0 : 60 c0 20 6d c0 ad 5d c0   d6
c1f8 : 8d 65 c0 a0 01 ad 62 c0   05
c200 : 8d 66 c0 8c 67 c0 c8 ad   7d
c208 : 63 c0 cd 66 c0 b0 06 8d   d0
c210 : 66 c0 8c 67 c0 c8 ad 64   b8
c218 : c0 cd 66 c0 b0 06 8d 66   af
c220 : c0 8c 67 c0 c8 ad 65 c0   29
c228 : cd 66 c0 b0 06 8d 66 c0   56
c230 : 8c 67 c0 ac 67 c0 c0 03   bb
c238 : b0 0a c0 01 d0 03 4c 4c   2d
c240 : c2 4c 66 c2 d0 03 4c 86   7e
c248 : c2 4c ac c2 a9 07 8d 69   90
c250 : c0 a9 03 8d 68 c0 20 cc   fe
c258 : c2 ce 68 c0 10 f8 ce 69   8a
c260 : c0 10 ee 4c 7b c3 a9 07   f8
c268 : 8d 69 c0 a9 00 8d 68 c0   9f
c270 : 20 cc c2 ac 68 c0 c8 8c   05
c278 : 68 c0 c0 04 90 f2 ce 69   a0
c280 : c0 10 e8 4c 7b c3 a9 00   88
c288 : 8d 69 c0 a9 00 8d 68 c0   bf
c290 : 20 cc c2 ac 68 c0 c8 8c   25
c298 : 68 c0 c0 04 90 f2 ac 69   37
c2a0 : c0 c8 8c 69 c0 c0 08 90   68
c2a8 : e2 4c 7b c3 a9 00 8d 69   ab
c2b0 : c0 a9 03 8d 68 c0 20 cc   5e
c2b8 : c2 ce 68 c0 10 f8 ac 69   62
c2c0 : c0 c8 8c 69 c0 c0 08 90   88
c2c8 : e8 4c 7b c3 ad 5e c0 0d   18
c2d0 : 68 c0 8d 60 c0 ad 5f c0   80
c2d8 : 0d 69 c0 8d 61 c0 20 6d   f3
c2e0 : c0 90 01 60 ad 6b c0 0d   88
c2e8 : 69 c0 85 f7 ad 6c c0 18   83
c2f0 : 69 40 85 f8 ad 68 c0 0a   2f
c2f8 : 0a 18 6d 5c c0 aa bd 2d   a8
c300 : c0 8d 6a c0 a0 00 b1 f7   fa
c308 : 0d 6a c0 91 f7 ad 6c c0   cd
c310 : 4a 85 fa ad 6b c0 6a 85   03
c318 : f9 46 fa 66 f9 46 fa 66   4a
c320 : f9 a5 fa 18 69 60 85 fa   53
c328 : a0 00 b1 f9 aa 4a 4a 4a   2f
c330 : 4a 85 fb 8a 29 0f 85 fc   a8
c338 : 18 a5 fa 69 04 85 fa b1   ca
c340 : f9 85 fd ae 5b c0 bd 40   84
c348 : 5f ae 5c c0 e0 02 90 05   98
c350 : d0 06 85 fc 2c 85 fb 2c   5b
c358 : 85 fd a5 fd 91 f9 38 a5   1a
c360 : fa e9 04 85 fa a5 fb 0a   e1
c368 : 0a 0a 0a 05 fc 91 f9 18   0f
c370 : a5 fa 69 78 85 fa a5 fd   be
c378 : 91 f9 60 ad 6b c0 18 69   c4
c380 : 08 8d 6b c0 90 03 ee 6c   f7
c388 : c0 ad 5e c0 18 69 04 8d   c6
c390 : 5e c0 c9 a0 b0 03 4c b4   92
c398 : c1 ad 5f c0 18 69 08 8d   28
c3a0 : 5f c0 c9 c8 b0 08 a9 00   dd
c3a8 : 8d 5e c0 4c b4 c1 58 60   99
c3b0 : 24 a9 02 a2 08 a0 00 20   44
c3b8 : ba ff a9 01 a2 b0 a0 c3   b6
c3c0 : 20 bd ff 20 c0 ff a2 02   5d
c3c8 : 20 c6 ff 20 cf ff 20 cf   6c
c3d0 : ff 20 cf ff a5 90 d0 32   5a
c3d8 : 20 cf ff f0 2d a6 90 d0   ea
c3e0 : 29 20 cf ff a6 90 d0 22   84
c3e8 : 85 fa 20 cf ff a6 90 d0   05
c3f0 : 19 a6 fa 20 cd bd 20 cf   0a
c3f8 : ff a6 90 d0 0d 20 d2 ff   a6
c400 : d0 f4 a9 0d 20 d2 ff 4c   88
c408 : d1 c3 20 cc ff a9 02 4c   4a
c410 : c3 ff 20 fd ae 20 d4 e1   9e
c418 : a9 00 a8 a2 40 20 d5 ff   9c
c420 : b0 0b 20 b7 ff 29 bf f0   7f
c428 : 03 4c 9c e1 60 4c f9 e0   c7
c430 : 20 fd ae 20 d4 e1 a2 00   e5
c438 : 8a a0 68 85 f7 a9 40 85   b6
c440 : f8 a9 f7 20 d8 ff b0 e5   2b
c448 : 60 a9 00 a8 85 f7 a9 40   d1
c450 : 85 f8 a2 1f a9 00 91 f7   af
c458 : c8 d0 fb e6 f8 ca d0 f6   7b
c460 : 91 f7 c8 c0 40 90 f9 60   68
c468 : a9 00 85 f7 85 f9 85 14   d8
c470 : a9 60 85 f8 a9 64 85 fa   93
c478 : a9 d8 85 15 a2 04 ad 65   5d
c480 : 5f 91 f7 ad 66 5f 91 f9   f7
c488 : 91 14 c8 d0 f1 e6 f8 e6   77
c490 : fa e6 15 ca d0 e8 60 00   72
c498 : 01 0f 0c 0b 07 02 0a 06   3a
c4a0 : 0e 03 09 08 05 0d 04 00   3c
c4a8 : 00 00 00 00 00 00 00 00   a9
c4b0 : 00 00 00 00 00 00 00 00   b1
c4b8 : 00 00 00 00 00 00 00 00   b9
c4c0 : ff ff ff ff a2 0f bd a2   9d
c4c8 : 0f bd 97 c4 9d 40 5f ca   23
c4d0 : 10 f7 60 a2 0e bd 41 5f   db
c4d8 : 9d a7 c4 ca 10 f7 ad 41   cd
c4e0 : 5f 8d b6 c4 a9 00 85 f7   ed
c4e8 : 85 f9 85 14 a8 aa a9 60   95
c4f0 : 85 f8 a9 64 85 fa a9 d8   71
c4f8 : 85 15 a2 0f bd 40 5f a8   3f
c500 : bd a7 c4 99 b7 c4 ca 10   e2
c508 : f3 a0 00 78 ad 11 d0 30   61
c510 : fb ad 12 d0 c9 36 90 f4   fb
c518 : b1 f7 aa 4a 4a 4a 4a a8   2a
c520 : b9 b7 c4 85 fb 8a 29 0f   6d
c528 : aa bd b7 c4 85 fc a0 00   fa
c530 : b1 f9 aa bd b7 c4 91 f9   1c
c538 : 91 14 a5 fb 0a 0a 0a 0a   e9
c540 : 05 fc 91 f7 e6 f7 e6 14   18
c548 : e6 f9 d0 06 e6 f8 e6 15   1c
c550 : e6 fa a5 f8 c9 63 b0 03   bc
c558 : 4c 18 c5 a5 f7 c9 e8 b0   a9
c560 : 03 4c 18 c5 58 a2 0f bd   9a
c568 : a7 c4 9d 40 5f ca 10 f7   5d
c570 : 60 a2 04 a0 00 98 85 f7   01
c578 : 85 f9 a9 64 85 f8 a9 d8   69
c580 : 85 fa b1 f7 91 f9 c8 d0   9b
c588 : f9 e6 f8 e6 fa ca d0 f2   3e
c590 : 60 a9 00 a8 85 f7 85 f9   fc
c598 : a9 60 85 f8 a9 68 85 fa   dc
c5a0 : a2 08 4c 82 c5 a9 00 a8   a5
c5a8 : 85 f7 85 f9 a9 68 85 f8   af
c5b0 : a9 60 85 fa a2 08 4c 82   eb
c5b8 : c5 ad 11 d0 09 20 8d 11   9c
c5c0 : d0 a9 81 8d 18 d0 a9 02   2a
c5c8 : 8d 00 dd ad 16 d0 09 10   af
c5d0 : 8d 16 d0 4c 71 c5 ad 16   4e
c5d8 : d0 29 ef 8d 16 d0 ad 11   ab
c5e0 : d0 29 df 8d 11 d0 a9 15   57
c5e8 : 8d 18 d0 a9 c7 8d 00 dd   8f
c5f0 : 4c 44 e5 a9 50 a0 5f 4c   2d
c5f8 : a2 bb a2 50 a0 5f 4c d4   0a
c600 : bb 00 ff ff ff ff 00 00   bb
```

Listing 2.
Diese Routine dient zur Berechnung der Grafiken. Bitte mit dem MSE (Seite 159) eingeben.


LANDSCHAFTEN AUS
Foto: Dieter Seibert

DEM HEIMCOMPUTER



Hier zeigt sich das Chaos von seiner besten Seite. Die grafische Darstellung der sogenannten Fractals auf dem C 64 erzeugt Zufalls-Landschaften, die der Natur recht ähnlich sind. Nicht umsonst werden diese Grafiken auch in Computerspielen (z. B. Rescue on Fractalus oder Koronis Rift) und professionellen Simulationen zur Landschaftserzeugung benutzt. Dort allerdings nicht mit den hier benutzten Zufallselementen, sondern mit geplanten und vorberechneten Stützpunkten.

Eingabehinweise

Tippen Sie das Basic-Listing »Fractale.bas 1.4« (Listing 1) sowie die MSE-Listings »Fractale 9.0« (Listing 2) »Biene/Maus« (Listing 3) und »Menüzeile« (Listing 4) ab und speichern Sie sie auf derselben Seite einer Diskette. (Dies ist wichtig, da das Basic-Programm die übrigen Programme nachlädt.)

Bedienungsanleitung

Haben Sie die Listings wie unter den Eingabehinweisen angegeben abgetippt und gespeichert, so können Sie mit

```
LOAD "FRACTALE.BAS*",8:RUN
```

das Hauptprogramm laden und starten. Es erscheint dann die Meldung »Bitte warten!«, denn nun werden die drei übrigen Programme nachgeladen. Danach meldet sich das Fractal-Programm mit der Ausgangsgrafik für eine Landschaft und seiner Menüzeile am oberen Bildschirmrand. Außerdem ist noch ein Maus-Cursor in Form eines weißen Pfeils sichtbar, den Sie mit dem Joystick in Port 2 bewegen können. Mit Hilfe dieses Cursors können Sie die folgenden Befehle der Menüzeile anwählen: NEXT, RUN, DLOAD, DSAVE, GLOAD, GSAVE.

Das Anwählen eines Befehls geschieht durch Plazieren des Pfeils auf dem jeweiligen Befehl und anschließendes zweimaliges Drücken des Feuerknopfes. Nach dem ersten Druck auf die Feuertaste wird der dadurch angewählte Befehl in der Menüleiste invertiert. Klickt man ihn nun noch einmal an, so wird er ausgeführt. Hat man den Pfeil inzwischen auf einen anderen Befehl bewegt, so wird dieser invertiert und der vorher gewählte Befehl wieder normal (schwarz-grau) dargestellt. Das gleiche geschieht auch, wenn Sie nach einmaligem Anklicken des Befehls den Pfeil-Cursor von der Menüleiste herunterbewegen und erneut den Feuerknopf betätigen.

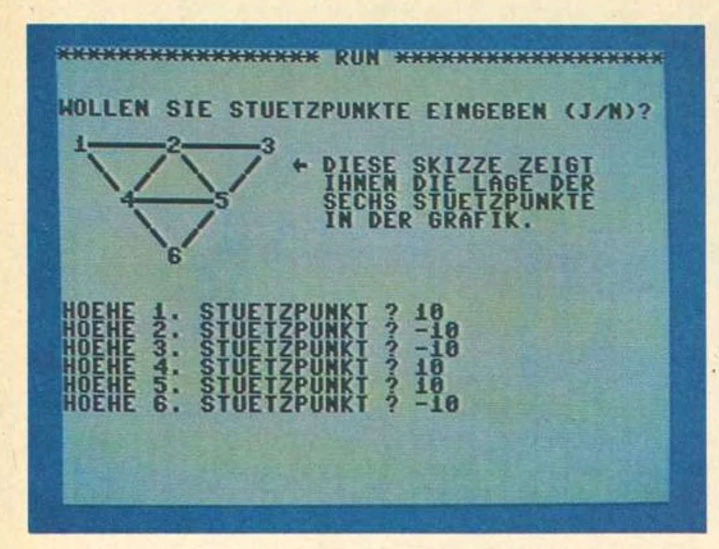
NEXT

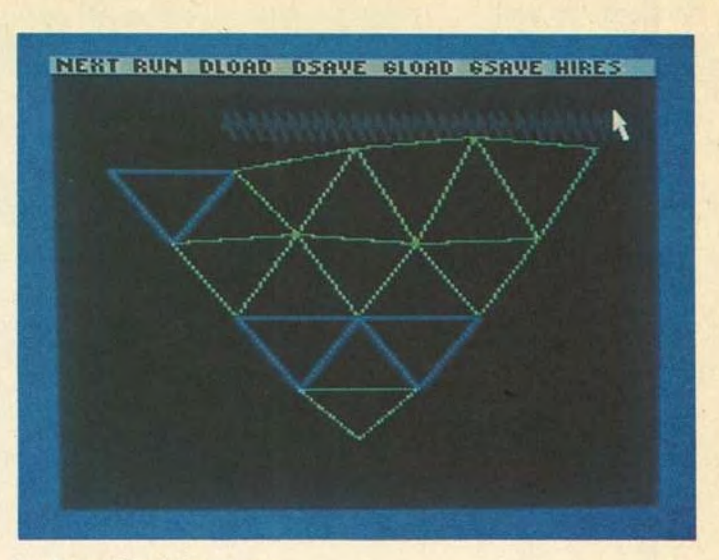
Dieser Befehl bewirkt, daß das Programm die nächste Verfeinerung der Fractal-Grafik berechnet, sofern dies möglich, d. h. noch sinnvoll ist. Bereits nach sechsmaligem Halbieren der Dreiecksseiten ist nämlich die Grenze des Auflösungsvermögens erreicht. Man sieht dies in der Grafik daran, daß kaum noch die Linien zu erkennen sind, aus denen sie aufgebaut ist, sondern diese zu unregelmäßigen Flächen zusammengerückt sind.

Wählt man in dieser Situation trotzdem NEXT an, so geschieht nichts; der C 64 führt diesen Befehl nicht aus. Man befindet sich danach weiterhin im Auswahlmodus.

Der Computer zeigt übrigens während des Berechnungsvorganges eine kleine Biene anstelle des Pfeils als Zeichen dafür, daß er sozusagen »emsig« beschäftigt ist. Dies gilt auch für alle anderen Programmpunkte, in denen

Auch Computer können kreativ sein. Ein gutes Beispiel dafür sind die Fractals, die richtig dargestellt der Natur ähnliche Gebirgszüge und Landschaften zeigen. Entwerfen Sie Ihre eigene Computer-Gegend, wie es sonst nur die Profis beherrschen.





der Computer beschäftigt ist, und man nicht sofort erkennen kann, daß er wirklich etwas tut und nicht abgestürzt ist.

Will man nun eine neue Landschaft erstellen, so muß man den nächsten Befehl anwählen:

RUN

Damit starten Sie das Maschinenprogramm neu, haben aber — im Gegensatz zum eigentlichen Programmstart vom Basic aus mit RUN — die Möglichkeit, die Lage der Ausgangsdreiecke für die nächste Grafik dadurch zu beeinflussen, daß Sie nun Stützpunkte eingeben können. Sie werden vom Programm gefragt:

Wollen Sie Stützpunkte eingeben (j/n)?

Wenn Sie diese Frage mit <N> wie »nein« beantworten, befindet sich das Programm einige Augenblicke später im gleichen Zustand wie nach dem Start, und Sie können erneut auf der Menüleiste auswählen. Der C 64 wählt die Stützpunkte danach über Zufallszahlen.

Haben Sie <J> für »ja« getippt, so erscheint nun eine Skizze auf dem Bildschirm, die Sie über die Lage der Stützpunkte auf dem Bildschirm aufklärt. Sodann werden Sie aufgefordert, die Höhen der sechs Stützpunkte einzugeben. Mit diesen Stützpunkten können Sie das Aussehen der Landschaft entscheidend mitbestimmen, da die Lage dieser Punkte im weiteren Programmverlauf nicht verändert wird.

Geben Sie z. B. folgende Werte für die Höhen ein:
10, -10, 10, -10, -10, 10

Sie erhalten dann mehr oder weniger Berge an den Ecken des großen Dreiecks und einen See in der Mitte.

Wählen Sie hingegen
-5, 10, -5, 10, 10, -5
so ergibt sich eine Insel in der Bildschirmmitte, die vom Meer umgeben ist. Experimentieren Sie ruhig mit den Werten, und lassen Sie sich von den Ergebnissen überraschen! Es dürfen allerdings nur Werte eingegeben werden, die zwischen -31 und 31 liegen.

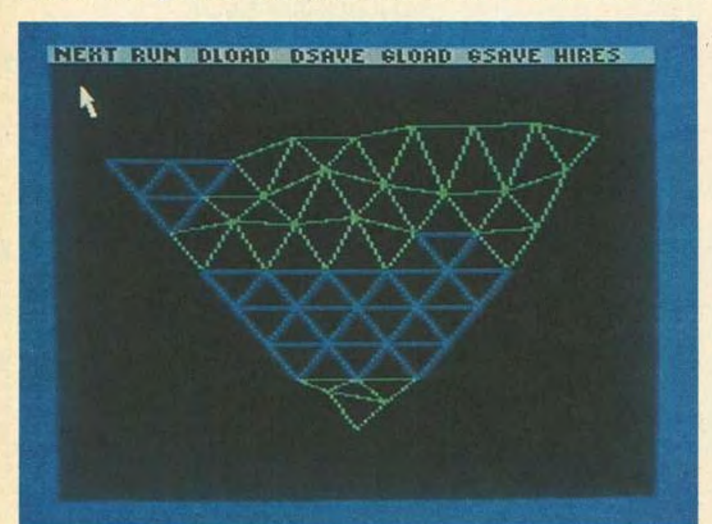
Bei der Eingabe von Stützpunkten können für jeden der sechs Punkte 61 verschiedene Werte eingegeben werden. Also sind nur mit Hilfe der Stützpunkte $61^6 = 5{,}2 * 10^{10}$ unterschiedliche Kombinationen und damit Landschaften möglich! Und dazu kommen dann noch die Veränderungen durch die nachfolgenden Rechenschritte.

DLOAD

Dieser Punkt dient dazu, die Höhendaten für eine Landschaft zu laden, die irgendwann vorher gespeichert wurden. Er dürfte vor allem dann Verwendung finden, wenn Sie ausprobieren wollen, was für unterschiedliche Grafiken aus derselben Ausgangslage entstehen können.

Haben Sie also diesen Befehl angewählt, so wird in den Textbildern umgeschaltet, und der Computer fragt Sie nach dem Namen des zu ladenden Files. Geben Sie hier den Namen ein, unter dem Sie es gespeichert hatten. Sollten Sie den Namen einmal nicht mehr wissen, so können Sie sich durch Eingabe von <$> als Filename das Inhaltsverzeichnis der gerade eingelegten Diskette ausgeben lassen.

Wollen Sie diesen Programmteil verlassen, ohne irgendwelche Daten zu laden, so gelangen Sie durch Eingabe eines <RETURN> als Filename zurück zur Grafik und in den Auswahlmodus.



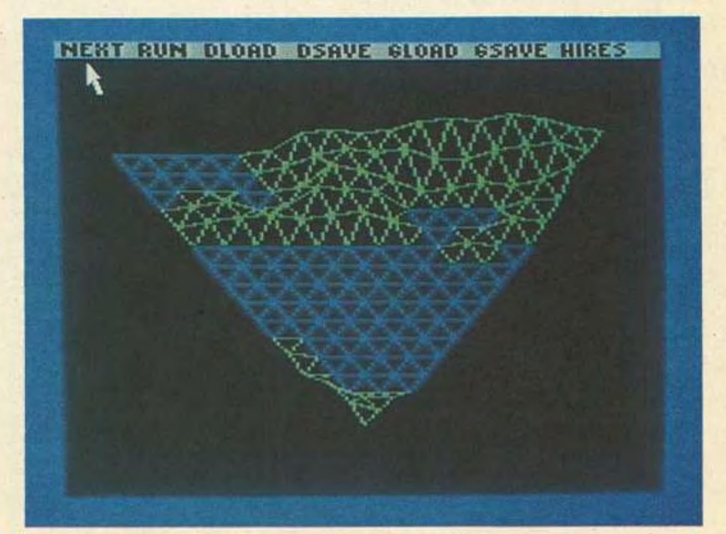


Bild 1 bis 6. Nach der Eingabe der Werte entwickelt sich Schritt für Schritt eine Landschaft mit Hügeln und Seen

Die Befehle <$> und <RETURN> können auch in den folgenden Programmpunkten angewandt werden.

Im Filenamen dürfen auch die sogenannten Jokerzeichen (»?«, »*«) vorkommen; die Eingabe eines »*« allein als Filename veranlaßt den Computer dazu, das erste Datenfile auf der Diskette zu laden. Gleiches gilt auch für den Programmpunkt GLOAD, hingegen nicht für die Punkte DSAVE und GSAVE, da Jokerzeichen beim Speichern nicht zulässig sind.

DSAVE

Hiermit lassen sich die Höhendaten der gerade dargestellten Landschaft speichern. Auch hier wird wieder in den Textmodus zurückgeschaltet, und das Programm fragt nach dem Namen, unter dem die Daten gespeichert werden sollen. Es empfiehlt sich, einen für die jeweilige Landschaft charakteristischen Namen zu verwenden, z. B. »Atolle«, »Inselgruppe«, »Fjorde«.

GLOAD

Mit diesem Befehl können Sie Grafiken, die einmal gespeichert wurden, wieder laden. Auch hier muß wieder der Name des zu ladenden Files eingegeben werden. Diese Grafik wird dann so lange gezeigt, bis Sie den Feuerknopf an Ihrem Joystick betätigen. Dann wird wieder die aktuelle Grafik, die sich vor dem Laden auf dem Bildschirm befand, eingeblendet. War diese im gleichen Modus (Multi/Hires) wie die geladene Grafik, so erfolgt ein weiches, zeilenweises Umblenden, das Sie auch bei der Ausführung des Befehls NEXT beobachten können. Handelte es sich hingegen um unterschiedliche Modi, so wird die Grafik gelöscht, der vorher herrschende Modus eingeschaltet und die alte Grafik dargestellt.

GSAVE

Dieser Befehl wurde implementiert, da der Benutzer sicherlich früher oder später eine besonders gelungene Landschaft zu irgendwelchen anderen Zwecken weiterverwenden oder mit einem Zeichenprogramm bearbeiten will.

Hier muß wieder der Name eingegeben werden, unter dem die Grafik auf Diskette gespeichert werden soll. Die Befehle <$> sowie <RETURN> sind auch hier möglich; die Verwendung von Jokerzeichen hingegen nicht.

HIRES/MULTI

Dieser Befehl ist der letzte in der Menüleiste. Je nach gerade aktivem Grafikmodus wird der entsprechende Befehl eingeblendet. Im Hires-Modus wird MULTI angezeigt, bei Multicolor HIRES.

Dieser Befehl dient dazu, die gerade in einem bestimmten Modus gezeigte Landschaft in dem anderen Modus

Kurzinfo: Computerlandschaften

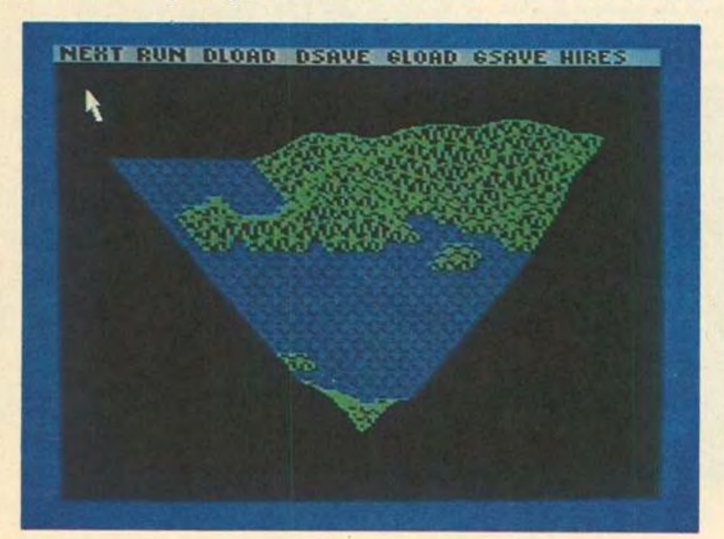
Programmart: Tool zum Generieren von »Computerlandschaften«
Laden: LOAD "FRACTALE.BAS*",8
Start: Nach dem Laden RUN eingeben
Besonderheiten: Das Basic-Programm lädt die Maschinensprache-Programme »FRACTALE 9.0«, »BIENE/MAUS« und »MENUEZEILE« nach. Die Steuerung erfolgt über einen Joystick in Port 2. Verändert sich der Cursor-Pfeil zu einer »Biene«, befindet sich der Computer in einer Berechnungsphase.
Programmautor: I. Camphausen

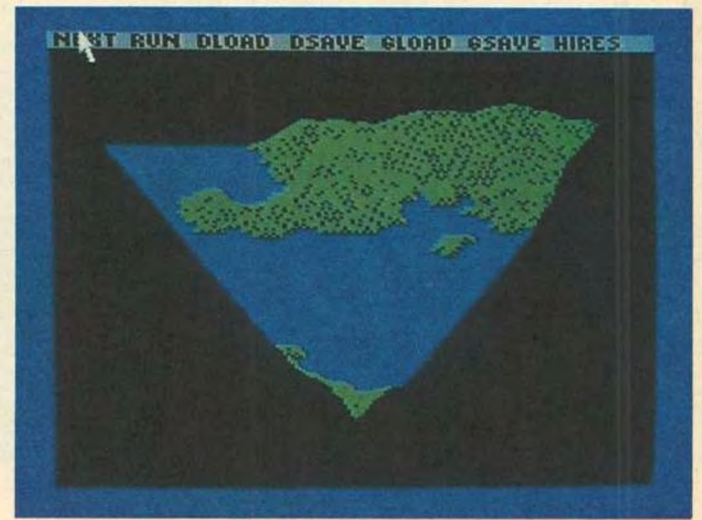
darzustellen. Dies ist allerdings nur möglich, wenn sich auch die zugehörigen Daten gerade im Speicher befinden, denn bei der Ausführung dieses Befehls muß die Landschaft neu gezeichnet werden. Er kann also nicht angewählt werden, wenn gerade eine mit GLOAD geladene Grafik angezeigt wird.

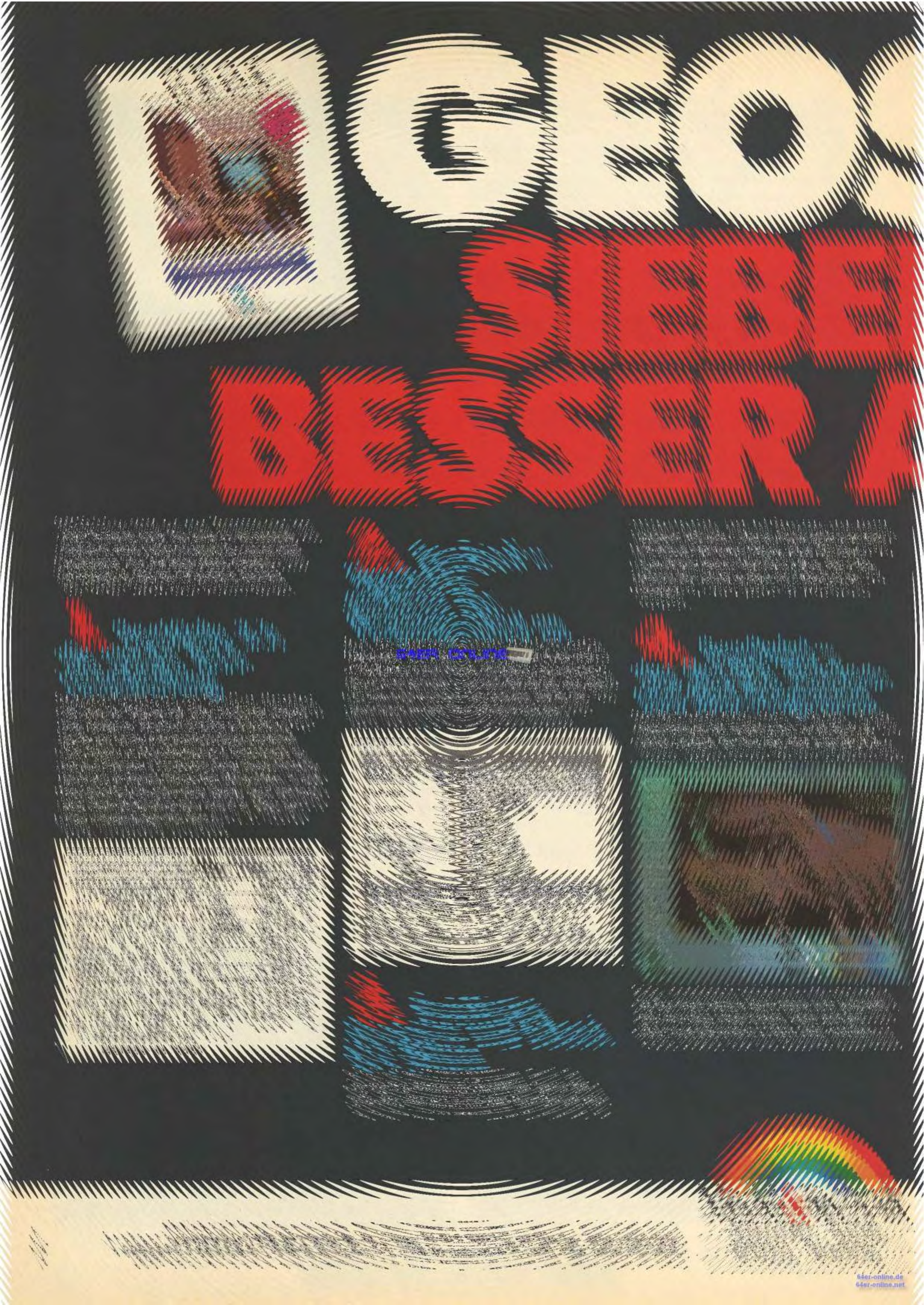
Im Multicolor-Modus werden die Landschaften mit grünen Bergen und blauen Gewässern vor schwarzem Hintergrund gezeichnet; im Hires-Modus nur grün auf schwarz, da hier nur zwei Farben möglich sind. Die Wasserflächen kann man daran erkennen, daß sie aus besonders regelmäßigen Dreiecken bestehen. Die Dreiecke sind im Multicolor-Modus nicht mehr zu erkennen.

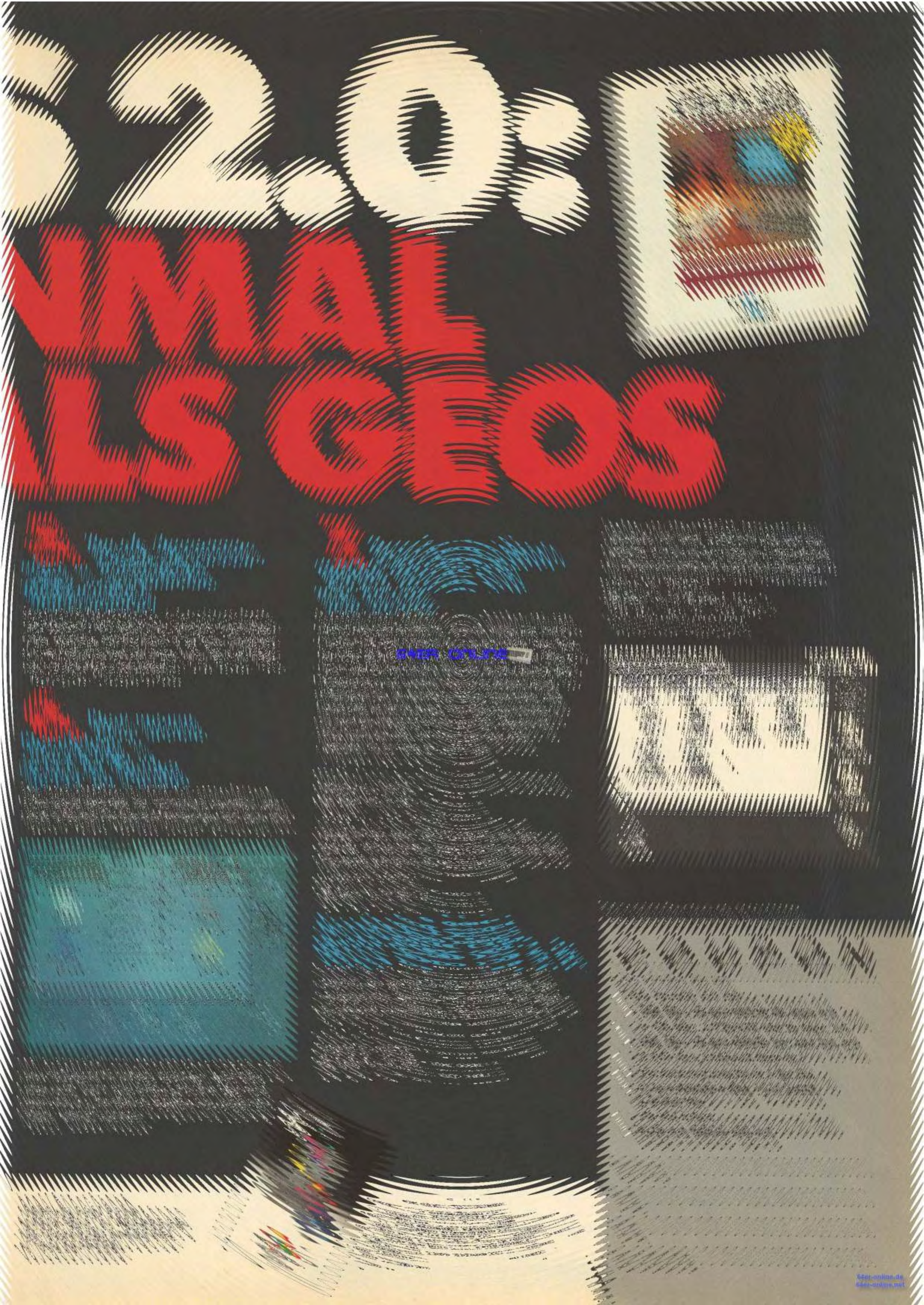
Im folgenden finden Sie eine Auflistung der Speicherstellen, deren Inhalte Sie je nach Wunsch verändern können, um das Programm nach Ihrem Gutdünken zu gestalten:

Rahmenfarbe
32774=$8006, 0 (Schwarz)...15 (Hellgrau)
Hintergrundfarbe für Text
34208=$85A0, 0...15
Farben für Grafik
a) Hires-Grafik:
34078=$851E Low-Nibble=Hintergrund
High-Nibble=Zeichenfarbe
b) Multicolor-Grafik
34075=$851B Low-Nibble=Farbe für Wasserflächen
High-Nibble=Farbe für Berge
33133=$816D Low-Nibble=Hintergrundfarbe
Farbe(n) der Menüzeilen
a) Hires:
34099=$8533 Low-Nibble=Zeichenfarbe
High-Nibble=Hintergrundfarbe
b) Multicolor:
34104=$8538 Low-Nibble=Hintergrundfarbe
Farbe für Maus/Biene
35142=$8946 Low-Nibble=Farbe der Maus
35131=$893B Low-Nibble=Farbe der Biene






GEOS
SIEBE
BESSER A



Geschwindigkeit der Maus
34989=$88AD, 1 (schnell)...255 (sehr langsam)
Mindestpause beim Doppelklick
32846=$804E, 1 (kurz)...255 (lang)
Verzögerung beim zeilenweisen Umblenden
35267=$89C3, 1 (gering)...255 (groß)

Experimentieren Sie mit dem Programm, werden Sie die Grafiken ebenso faszinierend finden wie wir. Sollten Sie Landschaften für eigene Programme verwenden, informieren Sie uns; interessante Programme sind immer willkommen. (I. Camphausen/ef)

```
10 REM FRACTALS (BASIC-STEUERPGM)            <076>
20 REM V1.4                                  <233>
30 REM FUER C64 UND VC1541                   <252>
40 REM GESCHRIEBEN VON                       <004>
50 REM I. CAMPHAUSEN                         <249>
60 :                                         <036>
70 :                                         <046>
80 :                                         <056>
100 IF A=0 THEN PRINT"{CLR,12DOWN}";TAB(13
    );"BITTE WARTEN!"                        <042>
110 IF A=0 THEN POKE 55,0:POKE 56,96:CLR:A
    =1:LOAD"FRACTALE 9.0",8,1                <118>
120 IF A=1 THEN A=2:LOAD"MENUEZEILE",8,1     <045>
130 IF A=2 THEN A=3:LOAD"BIENE/MAUS",8,1     <036>
140 :                                        <116>
400 REM                                      <208>
410 REM **************************           <196>
420 REM EINSPRUNGSADRESSEN FUER MC           <107>
430 REM **************************           <216>
440 REM                                      <248>
500 RU=32799                                 <076>
510 R2=32823                                 <171>
520 BEFEHL=35115                             <177>
530 GIN=32995                                <027>
540 TASTE=35601                              <237>
550 DIR=35617                                <218>
560 NLAENGE=35887                            <055>
570 DLOA=35888                               <088>
580 DSAV=35901                               <006>
590 GLOA=35924                               <066>
600 GSAV=35937                               <098>
610 :                                        <078>
620 A1(1)=DL:A1(2)=DS:A1(3)=GL:A1(4)=GS      <123>
630 A2(1)=RU:A2(2)=R2:A2(3)=GIN:A2(4)=R2     <058>
640 FOR I=1 TO 6:READ AD(I):NEXT I           <255>
650 DATA 24576,24608,24640,26656,26688,287
    36                                       <053>
660 FOR I=1 TO 4:READ UE$(I):NEXT I          <213>
670 DATA " DLOAD "," DSAVE "," GLOAD "," G
    SAVE "                                   <088>
680 :                                        <148>
690 :                                        <158>
1000 REM                                     <044>
1010 REM *************                       <010>
1020 REM HAUPTPROGRAMM                       <047>
1030 REM *************                       <032>
1040 REM                                     <086>
1050 SYS 32768                               <231>
1060 :                                       <020>
1070 NR=PEEK(BEFEHL)-2:IF NR>0 THEN GOSUB
     3000:GOTO 1070                          <082>
1080 GOSUB 2000:GOTO 1070                    <160>
1090 :                                       <050>
1100 :                                       <060>
2000 REM                                     <028>
2010 REM ************************            <029>
2020 REM UP STUETZPUNKTE EINGEBEN            <252>
2030 REM ************************            <049>
2040 REM                                     <068>
2050 PRINT:PRINT                             <226>
2060 FOR I=1 TO 6                            <001>
2070 :  H=0                                  <129>
2080 :  PRINT"HOEHE"I"{LEFT}. STUETZPUNKT
     ";:INPUT H                              <207>
2090 :  IF ABS(H)>30 THEN PRINT"{RVSON}BIT
     TE BEACHTEN: -30 <= HOEHE <= +30!{RVO
     FF}":GOTO 2080                          <109>
2100 :  IF H<0 THEN H=256+H                  <073>
2110 :  POKE AD(I),H                         <222>
2120 NEXT I                                  <172>
2130 SYS RU                                  <195>
2140 RETURN                                  <166>
2150 :                                       <094>
2160 :                                       <104>
3000 REM                                     <012>
3010 REM ******************                  <003>
3020 REM UP LADEN/SPEICHERN                  <046>
3030 REM ******************                  <023>
3040 REM                                     <052>
3050 PRINT "{CLR,BLACK}******************
     *********************"                  <059>
3060 PRINT "{HOME}";TAB((40-LEN(UE$(NR)))/
     2);UE$(NR)                              <087>
3070 N$="":INPUT"FILENAME ";N$               <237>
3080 IF N$="" THEN SYS R2:RETURN             <035>
3090 IF N$="$"THEN SYS DIR:GOSUB 4000:IF F
     E=0 THEN SYS TASTE:GOTO 3000            <243>
3100 IF NR<3 THEN N$=CHR$(4)+N$              <233>
3110 IF NR>2 THEN N$=CHR$(7)+N$              <211>
3120 POKE NL,LEN(N$)                         <159>
3130 FOR I=1 TO LEN(N$):POKE 703+I,ASC(MID
     $(N$,I,1)):NEXT I                       <019>
3140 SYS A1(NR):GOSUB 4000:IF FE<>0 THEN 3
     000                                     <022>
3150 SYS A2(NR)                              <247>
3160 RETURN                                  <170>
3170 :                                       <098>
3180 :                                       <108>
4000 REM                                     <252>
4010 REM ***********************             <085>
4020 REM UP FEHLERKANAL ABFRAGEN             <172>
4030 REM ***********************             <105>
4040 REM                                     <036>
4050 OPEN 1,8,15:INPUT#1,FE,FE$,A,B:CLOSE
     1                                       <161>
4060 PRINT:PRINT FE,FE$:IF FE<>0 THEN SYS
     TASTE                                   <039>
4070 RETURN                                  <062>
```

Listing 1. »Fractals.bas. 1.4« geben Sie bitte mit dem Checksummer ein

```
Name : fractale 9.0       8000 8c7e
-----------------------------------
8000 : a9 08 20 d2 ff a9 06 8d  90
8008 : 20 d0 20 d5 89 20 ed 89  b7
8010 : 20 90 81 20 3a 89 20 45  d8
8018 : 85 20 fb 84 20 ad 81 20  f2
8020 : fb 84 20 45 85 20 90 81  ad
8028 : 20 3a 89 20 15 85 20 39  3c
8030 : 83 20 b6 8b 20 45 89 20  75
8038 : 45 85 20 15 85 20 90 81  89
8040 : 20 4c 81 a9 00 8d 2b 89  48
8048 : a9 01 8d bf 88 a2 46 20  c4
8050 : b0 88 20 30 89 20 23 88  8a
8058 : ad c0 88 d0 e3 20 c3 88  01
8060 : cd 2b 89 f0 0c 8d 2b 89  30
8068 : 20 4c 81 20 d1 88 4c 48  36
8070 : 80 c9 01 d0 19 a9 05 cd  be
8078 : ff 70 90 c4 20 3a 89 20  a6
8080 : ad 81 20 39 83 20 50 89  ab
8088 : 20 45 89 4c 40 80 c9 02  6a
8090 : d0 33 20 04 8b 20 83 85  55
8098 : a9 ff a0 89 20 1e ab 20  7c
80a0 : 11 8b ad 77 02 c9 4e d0  1a
80a8 : 03 4c 0a 80 c9 4a d0 ef  76
80b0 : 20 d5 89 20 ed 89 a9 01  f5
80b8 : 8d ff 70 20 db 82 a9 54  86
80c0 : a0 8a 4c 1e ab c9 03 d0  33
80c8 : 06 20 04 8b 4c 83 85 c9  db
80d0 : 04 d0 06 20 04 8b 4c 83  97
80d8 : 85 c9 05 d0 3e 20 04 8b  a9
80e0 : 4c 83 85 78 ad f9 84 48  ab
80e8 : a2 35 86 01 ad 50 ff 8d  5f
80f0 : f9 84 a2 37 86 01 58 20  cd
80f8 : 15 85 20 45 85 20 23 88  78
8100 : 68 cd f9 84 8d f9 84 f0  fa
8108 : 03 20 fb 84 20 45 85 20  2d
8110 : 15 85 20 50 89 20 90 81  d9
8118 : 4c 40 80 c9 06 d0 06 20  1d
8120 : 04 8b 4c 83 85 c9 07 d0  d2
```

```
8128 : 17 20 3a 89 ad f9 84 49  5e
8130 : 01 8d f9 84 20 39 83 20  21
8138 : fb 84 20 7a 85 20 15 85  85
8140 : 20 4c 81 20 50 89 20 45  47
8148 : 89 4c 40 80 a9 ff a0 71  17
8150 : 85 5f 84 60 a9 40 a0 e1  95
8158 : 85 58 84 59 18 a5 5f 69  55
8160 : 41 85 5a a5 60 69 01 85  0f
8168 : 5b 20 c0 a3 a9 00 8d 21  8b
8170 : d0 ad f9 84 d0 19 a9 40  23
8178 : a0 73 85 8b 84 8c a9 08  08
8180 : a0 e1 85 8d 84 8e a0 27  b1
8188 : b1 8b 91 8d 88 10 f9 60  c7
8190 : a9 b4 8d 00 d0 a9 3c 8d  5d
8198 : 01 d0 a9 01 8d 15 d0 8d  6c
81a0 : 27 d0 a9 00 8d 10 d0 a9  8a
81a8 : 2f 8d f8 cf 60 ee ff 70  34
81b0 : a9 00 8d 37 83 a9 80 8d  46
81b8 : 36 83 ad 36 83 49 ff 8d  80
81c0 : 36 83 30 5b ad fe 70 8d  df
81c8 : 38 83 38 ad 38 83 ed fd  d9
81d0 : 70 aa ac 37 83 18 20 e5  ed
81d8 : 82 85 61 18 ad 38 83 6d  fe
81e0 : fd 70 aa ac 37 83 18 20  86
81e8 : e5 82 18 65 61 18 10 01  da
81f0 : 38 6a 85 61 ad ff 70 20  c8
81f8 : 0a 83 18 65 61 ae 38 83  ea
8200 : ac 37 83 38 20 e5 82 18  9b
8208 : ad 38 83 6d fc 70 8d 38  5a
8210 : 83 18 ad 38 83 6d 37 83  99
8218 : c9 40 90 ae 4c ca 82 a9  74
8220 : 00 8d 38 83 38 ad 37 83  3a
8228 : ed fd 70 a8 ae 38 83 18  30
8230 : 20 e5 82 85 61 ae 38 83  08
8238 : 18 ad 37 83 6d fd 70 a8  3f
8240 : 18 20 e5 82 18 65 61 18  94
8248 : 10 01 38 6a 85 61 ad ff  4e
8250 : 70 20 0a 83 18 65 61 ae  53
8258 : 38 83 ac 37 83 38 20 e5  aa
8260 : 82 18 ad 38 83 6d fd 70  dd
8268 : 8d 38 83 38 ad 38 83 ed  80
8270 : fd 70 aa 18 ad 37 83 6d  d0
8278 : fd 70 a8 18 20 e5 82 85  21
8280 : 61 18 ad 38 83 6d fd 70  dc
8288 : aa 38 ad 37 83 ed fd 70  21
8290 : a8 18 20 e5 82 18 65 61  4a
8298 : 18 10 01 38 6a 85 61 ad  b3
82a0 : ff 70 20 0a 83 18 65 61  72
82a8 : ae 38 83 ac 37 83 38 20  99
82b0 : e5 82 18 ad 38 83 6d fd  e3
82b8 : 70 8d 38 83 38 a9 40 ed  1b
82c0 : 37 83 cd 38 83 90 03 4c  95
82c8 : 24 82 18 ad 37 83 6d fd  2a
82d0 : 70 8d 37 83 c9 40 b0 03  ac
82d8 : 4c ba 81 4e fc 70 4e fd  34
82e0 : 70 4e fe 70 60 08 48 a9  00
82e8 : bf 85 8b a9 5f 85 8c 18  06
82f0 : a5 8b 69 41 85 8b a5 8c  42
82f8 : 69 00 85 8c 88 10 f0 8a  36
8300 : a8 68 28 90 02 91 8b b1  37
8308 : 8b 60 a2 80 8e 18 d4 a2  be
8310 : ff 8e 0e d4 8e 0f d4 a2  6e
8318 : 00 8e 12 d4 a2 81 8e 12  13
8320 : d4 a8 88 b9 2f 83 2d 1b  9c
8328 : d4 c8 38 f9 2f 83 60 3f  bd
8330 : 1f 0f 07 03 01 00 00 00  09
8338 : 00 20 ff 84 a9 40 ae ff  30
8340 : 70 4a ca d0 fc 8d f8 84  cb
8348 : a9 00 8d ee 84 a9 00 8d  e3
8350 : ed 84 a9 02 8d fa 84 ad  48
8358 : ed 84 0a 18 6d ee 84 8d  88
8360 : ef 84 a9 00 69 00 8d f0  aa
8368 : 84 0e ef 84 2e f0 84 ad  58
8370 : ee 84 0a 18 69 32 8d f1  68
8378 : 84 ae ed 84 ac ee 84 18  e4
8380 : 20 e5 82 30 12 f0 10 85  2d
8388 : 61 38 ad f1 84 e5 61 8d  c7
8390 : f1 84 a9 01 8d fa 84 ad  6c
8398 : f8 84 0a 0a 18 6d ef 84  4c
83a0 : 8d f2 84 a9 00 69 00 8d  63
83a8 : f3 84 ad ee 84 0a 18 69  f2
83b0 : 32 8d f4 84 18 ad ed 84  26
83b8 : 6d f8 84 aa ac ee 84 18  9c
83c0 : 20 e5 82 30 12 f0 10 85  6d
83c8 : 61 38 ad f4 84 e5 61 8d  67
83d0 : f4 84 a9 01 8d fa 84 ad  af
83d8 : f8 84 0a 18 6d ef 84 8d  1b
83e0 : f5 84 a9 00 69 00 8d f6  3c
83e8 : 84 18 ad ee 84 6d f8 84  62
83f0 : 0a 18 69 32 8d f7 84 18  82
83f8 : ad ee 84 6d f8 84 a8 ae  9f
8400 : ed 84 18 20 e5 82 30 12  91
8408 : f0 10 85 61 38 ad f7 84  68
8410 : e5 61 8d f7 84 a9 01 8d  bd
8418 : fa 84 ad ef 84 18 69 1e  a9
8420 : 8d ef 84 ad f0 84 69 00  54
8428 : 8d f0 84 18 ad f2 84 69  a9
8430 : 1e 8d f2 84 ad f3 84 69  c1
8438 : 00 8d f3 84 18 ad f5 84  5c
8440 : 69 1e 8d f5 84 ad f6 84  75
8448 : 69 00 8d f6 84 ad ef 84  72
8450 : ae f0 84 ac f1 84 8d 10  c7
8458 : 88 8e 11 88 8c 12 88 ad  54
8460 : f2 84 ae f3 84 ac f4 84  49
8468 : 8d 13 88 8e 14 88 8c 15  55
8470 : 88 20 56 86 ad f2 84 ae  51
8478 : f3 84 ac f4 84 8d 10 88  7d
8480 : 8e 11 88 8c 12 88 ad f5  52
8488 : 84 ae f6 84 ac f7 84 8d  69
8490 : 13 88 8e 14 88 8c 15 88  60
8498 : 20 56 86 ad f5 84 ae f6  67
84a0 : 84 ac f7 84 8d 10 88 8e  a1
84a8 : 11 88 8c 12 88 ad ef 84  21
84b0 : ae f0 84 ac f1 84 8d 13  2d
84b8 : 88 8e 14 88 8c 15 88 20  71
84c0 : 56 86 18 ad ed 84 6d f8  c0
84c8 : 84 8d ed 84 a9 3f 38 ed  70
84d0 : ee 84 cd ed 84 90 03 4c  a3
84d8 : 52 83 ad ee 84 6d f8 84  d6
84e0 : 8d ee 84 c9 3f f0 02 b0  23
84e8 : 03 4c 4d 83 60 00 00 00  db
84f0 : 00 00 00 00 00 00 00 00  f1
84f8 : 00 01 02 a9 e0 d0 02 a9  1e
8500 : a0 85 fb a9 00 85 fa a8  00
8508 : a2 20 91 fa 88 d0 fb e6  4b
8510 : fb ca d0 f6 60 ad f9 84  e8
8518 : f0 03 a9 56 2c a9 50 a0  52
8520 : 00 99 00 cc 99 00 cd 99  8a
8528 : 00 ce 99 f8 ce 88 d0 f1  6d
8530 : a0 27 a9 f0 99 00 cc a9  0c
8538 : 0f 99 00 d8 88 10 f3 a9  5b
8540 : 00 8d 21 d0 60 ad 00 dd  98
8548 : 29 fc 8d 00 dd ad 02 dd  62
8550 : 09 03 8d 02 dd a9 38 8d  a6
8558 : 18 d0 a9 cc 8d 88 02 ad  5d
8560 : 11 d0 09 20 8d 11 d0 a9  18
8568 : ef 2d 16 d0 ae f9 84 f0  3c
8570 : 05 a9 10 0d 16 d0 8d 16  3a
8578 : d0 60 ad 16 d0 49 10 8d  59
8580 : 16 d0 60 a9 04 8d 88 02  1e
8588 : a9 97 8d 00 dd a9 15 8d  fb
8590 : 18 d0 a9 1b 8d 11 d0 ad  de
8598 : 16 d0 29 ef 8d 16 d0 a9  7f
85a0 : 0f 8d 21 d0 60 a9 80 30  8e
85a8 : 02 a9 00 8d 22 88 a6 03  38
85b0 : e0 c8 90 03 4c 1b 86 a5  7c
85b8 : 05 a4 04 c9 01 90 06 d0  98
85c0 : f3 c0 40 b0 ef 8a 4a 4a  4a
85c8 : 4a 0a a8 b9 1c 86 85 f8  77
85d0 : b9 1d 86 85 f9 8a 29 07  11
85d8 : 18 65 f8 85 f8 a5 04 48  ef
85e0 : 29 f8 18 69 00 48 a5 05  9b
85e8 : 69 a0 a8 18 68 65 f8 85  6f
85f0 : 61 98 65 f9 85 62 68 29  95
85f8 : 07 49 07 aa bd 4e 86 a0  65
8600 : 00 a2 35 78 86 01 2c 22  13
8608 : 88 10 07 49 ff 31 61 4c  2b
8610 : 14 86 11 61 a2 37 86 01  d8
8618 : 58 91 61 60 00 00 40 01  a0
8620 : 80 02 c0 03 00 05 40 06  67
8628 : 80 07 c0 08 00 0a 40 0b  c4
8630 : 80 0c c0 0d 00 0f 40 10  22
8638 : 80 11 c0 12 00 14 40 15  7f
8640 : 80 16 c0 17 00 19 40 1a  dc
8648 : 80 1b c0 1c 00 1e 01 02  02
8650 : 04 08 10 20 40 80 a0 01  ed
8658 : 8c 16 88 8c 17 88 8c 20  cb
8660 : 88 88 8c 18 88 8c 19 88  b5
8668 : 8c 21 88 88 ad 14 88 cd  f1
8670 : 11 88 90 0a d0 21 ad 13  1e
8678 : 88 cd 10 88 b0 19 38 ad  0c
8680 : 10 88 ed 13 88 8d 1a 88  21
8688 : ad 11 88 ed 14 88 8d 1b  90
8690 : 88 8c 16 88 4c aa 86 38  9a
8698 : ad 13 88 ed 10 88 8d 1a  5e
86a0 : 88 ad 14 88 ed 11 88 8d  ba
86a8 : 1b 88 ad 15 88 cd 12 88  66
86b0 : b0 0d 38 ad 12 88 ed 15  f2
86b8 : 88 8c 17 88 4c c3 86 38  cb
86c0 : ed 12 88 8d 1c 88 ad 1b  7d
86c8 : 88 d0 27 ad 1a 88 cd 1c  8d
86d0 : 88 b0 1f ae 1a 88 ad 1c  23
86d8 : 88 8e 1c 88 8d 1a 88 ad  e7
86e0 : 16 88 8d 19 88 ad 17 88  24
86e8 : 8d 18 88 c8 8c 16 88 8c  71
86f0 : 17 88 ad 1b 88 4a 8d 1f  69
86f8 : 88 ad 1a 88 6a 8d 1e 88  8b
8700 : 4c b9 87 ad 16 88 30 12  4b
8708 : 18 6d 10 88 8d 10 88 ad  c3
8710 : 11 88 69 00 8d 11 88 4c  dc
8718 : 2b 87 38 ad 10 88 e9 01  ba
8720 : 8d 10 88 ad 11 88 e9 00  8a
8728 : 8d 11 88 ad 18 88 30 07  aa
8730 : 18 6d 12 88 4c 3d 87 38  d2
8738 : ad 12 88 e9 01 8d 12 88  23
8740 : ee 20 88 d0 03 ee 21 88  b8
8748 : 18 ad 1e 88 6d 1c 88 8d  c4
8750 : 1e 88 ad 1f 88 69 00 8d  f1
8758 : 1f 88 ad 1f 88 cd 1b 88  7f
8760 : 90 57 d0 08 ad 1a 88 cd  3a
8768 : 1e 88 b0 4d 38 ad 1e 88  1b
8770 : ed 1a 88 8d 1e 88 ad 1f  59
8778 : 88 ed 1b 88 8d 1f 88 ad  1e
8780 : 19 88 30 0f 18 6d 10 88  0a
8788 : 8d 10 88 ad 11 88 69 00  f0
8790 : 4c a1 87 38 ad 10 88 e9  e7
8798 : 01 8d 10 88 ad 11 88 e9  ce
87a0 : 00 8d 11 88 ad 17 88 30  d2
87a8 : 07 18 6d 12 88 4c b6 87  2e
87b0 : 38 ad 12 88 e9 01 8d 12  55
87b8 : 88 ac 12 88 84 03 ae 10  67
87c0 : 88 ad 11 88 86 04 85 05  1d
87c8 : ad f9 84 f0 12 a5 04 29  62
87d0 : fe 85 04 ad fa 84 4a 90  66
87d8 : 06 20 a5 85 4c e2 87 20  43
87e0 : a9 85 ad f9 84 f0 15 e6  e8
```

Listing 2. »Fractale 9.0« geben Sie bitte mit dem MSE ein

```
87e8 : 04 d0 02 e6 05 ad fa 84   64
87f0 : 4a 90 06 20 a9 85 4c fc   fa
87f8 : 87 20 a5 85 ad 21 88 cd   4b
8800 : 1b 88 90 09 ad 1a 88 cd   0e
8808 : 20 88 b0 01 60 4c 03 87   3c
8810 : 00 00 00 00 00 00 00 00   11
8818 : 00 00 00 00 00 00 00 00   19
8820 : 00 00 00 78 ad 00 dc 58   2e
8828 : 4a aa b0 0a ad 01 d0 c9   ee
8830 : 33 90 03 ce 01 d0 8a 4a   9b
8838 : aa b0 0a ad 01 d0 c9 eb   08
8840 : b0 03 ee 01 d0 8a 4a aa   2d
8848 : b0 19 ad 10 d0 29 01 d0   ee
8850 : 07 ad 00 d0 c9 15 90 0b   e6
8858 : ce 00 d0 d0 06 ce 10 d0   2d
8860 : ce 00 d0 8a 4a aa b0 19   a3
8868 : ad 10 d0 29 01 f0 07 ad   86
8870 : 00 d0 c9 42 b0 0b ee 00   b2
8878 : d0 d0 06 ee 10 d0 ee 00   53
8880 : d0 a0 00 8a 4a b0 02 a0   65
8888 : 01 8c bf 88 ad 00 d0 38   5f
8890 : e9 14 48 ad 10 d0 e9 00   7a
8898 : 4a 68 6a 4a 4a 8d c1 88   23
88a0 : ad 01 d0 38 e9 32 4a 4a   f7
88a8 : 4a 8d c0 88 a2 02 a0 00   b7
88b0 : 88 d0 fd ca d0 fa ad bf   94
88b8 : 88 d0 03 4c 23 88 60 00   eb
88c0 : 00 00 00 ad c1 88 a0 00   59
88c8 : d9 23 89 c8 b0 fa 88 98   e4
88d0 : 60 ac 2b 89 b9 22 89 8d   70
88d8 : 2c 89 b9 23 89 8d 2e 89   6c
88e0 : 38 ad 2e 89 ed 2c 89 85   1d
88e8 : 61 a9 00 85 62 8d 2d 89   29
88f0 : a0 03 0e 2c 89 2e 2d 89   ed
88f8 : 06 61 26 62 88 d0 f3 ad   bf
8900 : 2c 89 85 8b ad 2d 89 18   5e
8908 : 69 e0 85 8c 78 a9 35 85   89
8910 : 01 a4 61 88 b1 8b 49 ff   69
8918 : 91 8b 88 10 f7 a9 37 85   48
8920 : 01 58 60 00 05 09 0f 15   64
8928 : 1b 21 26 00 00 00 05 00   72
8930 : 78 ad 00 dc 58 29 10 f0   0b
8938 : f7 60 a9 01 8d 27 d0 a9   92
8940 : 2e 8d f8 cf 60 a9 01 8d   df
8948 : 27 d0 a9 2f 8d f8 cf 60   c8
8950 : a9 35 78 85 01 a9 08 85   eb
8958 : 61 a5 61 48 4a 4a 4a 0a   21
8960 : a8 b9 1c 86 85 8b 85 8d   a3
8968 : b9 1d 86 85 8c 85 8e 18   61
8970 : a5 8c 69 a0 85 8c 18 a5   32
8978 : 8e 69 e0 85 8e 68 29 07   82
8980 : 48 18 65 8b 85 8b a5 8c   03
8988 : 69 00 85 8c 68 18 65 8d   dc
8990 : 85 8d a5 8e 69 00 85 8e   e1
8998 : a2 27 a0 00 b1 8b 91 8d   cf
89a0 : 18 a5 8b 69 08 85 8b a5   c1
89a8 : 8c 69 00 85 8c 18 a5 8d   d5
89b0 : 69 08 85 8d a5 8e 69 00   a5
89b8 : 85 8e ca 10 df a0 01 8c   59
89c0 : bf 88 a2 0a 20 b0 88 e6   25
89c8 : 61 a5 61 c9 c8 d0 8a a9   1e
89d0 : 37 85 01 58 60 a9 00 a0   aa
89d8 : 60 85 8b 84 8c a2 11 a0   d2
89e0 : 00 a8 91 8b 88 d0 fb e6   d7
89e8 : 8c ca d0 f6 60 a9 00 8d   5b
89f0 : ff 70 a9 20 8d fe 70 8d   43
89f8 : fd 70 0a 8d fc 70 60 93   5e
8a00 : 90 2a 2a 2a 2a 2a 2a 2a   66
8a08 : 2a 2a 2a 2a 2a 2a 2a 2a   08
8a10 : 2a 2a 20 52 55 4e 20 2a   3e
8a18 : 2a 2a 2a 2a 2a 2a 2a 2a   18
8a20 : 2a 2a 2a 2a 2a 2a 2a 2a   20
8a28 : 2a 0d 11 57 4f 4c 4c 45   1b
8a30 : 4e 20 53 49 45 20 53 54   d8
8a38 : 55 45 54 5a 50 55 4e 4b   10
8a40 : 54 45 20 45 49 4e 47 45   96
8a48 : 42 45 4e 20 28 4a 2f 4e   f3
8a50 : 29 3f 0d 00 11 20 31 c0   b5
8a58 : c0 c0 c0 c0 32 c0 c0 c0   6e
8a60 : c0 c0 33 0d 20 20 cd 20   69
8a68 : 20 20 ce 20 cd 20 20 20   ef
8a70 : ce 20 20 5f 20 44 49 45   16
8a78 : 53 45 20 53 4b 49 5a 5a   fd
8a80 : 45 20 5a 45 49 47 54 0d   4f
8a88 : 20 20 20 cd 20 ce 20 20   b3
8a90 : 20 cd 20 ce 20 20 20 20   3d
8a98 : 20 49 48 4e 45 4e 20 44   09
8aa0 : 49 45 20 4c 41 47 45 20   c1
8aa8 : 44 45 52 0d 20 20 20 20   89
8ab0 : 34 c0 c0 c0 c0 c0 35 20   b4
8ab8 : 20 20 20 20 20 53 45 43   2d
8ac0 : 48 53 20 53 54 55 45 54   d2
8ac8 : 5a 50 55 4e 4b 54 45 0d   f0
8ad0 : 20 20 20 20 20 cd 20 20   3e
8ad8 : 20 ce 20 20 20 20 20 20   2f
8ae0 : 20 49 4e 20 44 45 52 20   34
8ae8 : 47 52 41 46 49 4b 2e 0d   33
8af0 : 20 20 20 20 20 20 cd 20   a7
8af8 : ce 0d 20 20 20 20 20 20   1d
8b00 : 20 36 0d 00 a9 00 8d 15   7a
8b08 : d0 60 a9 a4 a0 8b 20 1e   2a
8b10 : ab a9 00 85 c6 a5 c6 f0   d7
8b18 : fc ad 77 02 a2 00 86 c6   db
8b20 : 60 a9 24 85 8b a9 8b 85   4e
8b28 : bb a9 00 85 bc a9 01 85   91
8b30 : b7 a9 08 85 ba a9 60 85   f4
8b38 : b9 20 d5 f3 a5 ba 20 b4   0f
8b40 : ff a5 b9 20 96 ff a9 00   94
8b48 : 85 90 a0 03 84 8b 20 a5   0e
8b50 : ff 85 8c a4 90 d0 49 20   be
8b58 : a5 ff a4 90 d0 42 a4 8b   01
8b60 : 88 d0 e9 a6 8c 20 cd bd   1c
8b68 : a9 20 20 d2 ff 20 a5 ff   1b
8b70 : a6 90 d0 2c aa f0 06 20   a2
8b78 : d2 ff 4c 6d 8b a9 0d 20   85
8b80 : d2 ff a5 d6 c9 16 90 14   4e
8b88 : 20 0a 8b a2 18 20 ff e9   3b
8b90 : ca e0 04 d0 f8 a9 b0 a0   c6
8b98 : 8b 20 1e ab a0 02 d0 ac   e7
8ba0 : 20 42 f6 60 0d 20 20 12   21
8ba8 : 54 41 53 54 45 92 0d 00   19
8bb0 : 13 11 11 11 11 00 a9 36   d7
8bb8 : 78 85 01 a9 a0 85 8c a9   24
8bc0 : e0 85 8e a9 00 85 8b 85   a1
8bc8 : 8d a8 a2 20 b1 8b 91 8d   2f
8bd0 : 88 d0 f9 e6 8c e6 8e ca   eb
8bd8 : d0 f2 a9 37 85 01 58 60   f5
8be0 : a2 08 a0 00 20 ba ff a2   cc
8be8 : c0 a0 02 ad 2f 8c 20 bd   82
8bf0 : ff a9 00 ae 2b 8c ac 2c   bc
8bf8 : 8c 4c d5 ff a9 36 78 85   59
8c00 : 01 a2 08 20 ba ff a2 c0   10
8c08 : a0 02 ad 2f 8c 20 bd ff   bb
8c10 : ae 2b 8c ac 2c 8c 86 8b   65
8c18 : 84 8c a9 8b ae 2d 8c ac   9e
8c20 : 2e 8c 20 d8 ff a9 37 85   ed
8c28 : 01 58 60 00 00 00 00 00   6e
8c30 : a9 00 8d 2b 8c a9 60 8d   55
8c38 : 2c 8c 4c e0 8b a9 00 8d   fb
8c40 : 2b 8c a9 60 8d 2c 8c a9   e7
8c48 : 00 8d 2d 8c a9 71 8d 2e   a4
8c50 : 8c 4c fc 8b a9 00 8d 2b   da
8c58 : 8c a9 e0 8d 2c 8c 4c e0   bd
8c60 : 8b a9 00 8d 2b 8c a9 a0   71
8c68 : 8d 2c 8c a9 00 8d 2d 8c   9e
8c70 : a9 c0 8d 2e 8c ad f9 84   ca
8c78 : 8d 50 bf 4c fc 8b ff ff   d3
```

Listing 2. »Fractale 9.0« (Schluß)

```
Name : biene/maus            cb80 cc00
---------------------------------------
cb80 : 00 00 e0 00 03 10 00 04   71
cb88 : 10 00 c8 10 01 28 20 07   ac
cb90 : 30 40 09 e1 80 09 1e 00   28
cb98 : 07 3c 00 03 f2 00 0c e7   4d
cba0 : 00 30 8e 80 40 fc 80 81   5d
cba8 : 39 80 82 13 80 8c 0f 00   cd
cbb0 : 70 00 00 00 00 00 00 00   21
cbb8 : 00 00 00 00 00 00 00 00   b9
cbc0 : 04 00 00 06 00 00 07 00   a1
cbc8 : 00 07 80 00 07 c0 00 07   f0
cbd0 : e0 00 07 f0 00 07 c0 00   cb
cbd8 : 06 e0 00 04 e0 00 00 70   be
cbe0 : 00 00 70 00 00 38 00 00   be
cbe8 : 38 00 00 00 00 00 00 00   21
cbf0 : 00 00 00 00 00 00 00 00   f1
cbf8 : 00 00 00 00 00 00 00 2b   4f
```

Listing 3. Der Grafik-Cursor für das Fractal-Programm

```
Name : menuezeile            7200 7368
---------------------------------------
7200 : ff f3 f0 f0 f3 f3 f3 ff   02
7208 : ff cc cc 0c 0c cc cc ff   7c
7210 : ff 0c fc 3f fc fc 0c ff   24
7218 : ff cc cf 3f cf cf cf ff   14
7220 : ff 0f 3f 3f 3f 3f 3f ff   4a
7228 : ff f0 f3 f3 f0 f3 f3 ff   99
7230 : ff f3 33 33 f3 33 3c ff   26
7238 : ff 33 30 30 33 33 f3 ff   80
7240 : ff cf cf 0f 0f cf cf ff   ab
7248 : ff fc fc fc fc fc fc ff   50
7250 : ff 3c cc cc cc cc 3c ff   5e
7258 : ff fc f3 f3 f3 f3 3c ff   20
7260 : ff fc 33 33 30 33 f3 ff   7d
7268 : ff f0 33 33 33 33 30 ff   a0
7270 : ff ff 3f 3f 3f 3f ff ff   15
7278 : ff fc fc fc fc fc fc ff   80
7280 : ff 3f cc cf cf cf 3c ff   b8
7288 : ff 0f fc 3c cc cc 3c ff   fa
7290 : ff 3c cc cc 0c cf cf ff   f8
7298 : ff cc cc cc cc 3c 3c ff   6a
72a0 : ff 0f ff 3f ff ff 0f ff   4b
72a8 : ff ff fc fc fc fc ff ff   3d
72b0 : ff 3c cc fc 0c cc 3c ff   b8
72b8 : ff fc f3 f3 f3 f3 3c ff   80
72c0 : ff fc 33 33 30 33 f3 ff   dd
72c8 : ff f0 33 33 33 33 30 ff   00
72d0 : ff ff 3f 3f 3f 3f ff ff   75
72d8 : ff fc f3 f3 f0 f3 fc ff   73
72e0 : ff fc 33 fc 3f 3f f0 ff   7b
72e8 : ff 3c f3 f3 30 33 f3 ff   ed
72f0 : ff f3 33 33 33 3c 3c ff   22
72f8 : ff 30 33 30 33 f3 f0 ff   79
7300 : ff 3f ff ff ff ff 3f ff   9c
7308 : ff cc cc cc c0 cc cc ff   e0
7310 : ff cc cc cc cc cc cc ff   a8
7318 : ff 3c cc cc 3c cc cc ff   5f
7320 : ff 0f fc 3f ff ff 0c ff   fe
7328 : ff 0f ff 3f cf cf 3f ff   0f
7330 : ff ff ff ff ff ff ff ff   2f
7338 : ff ff ff ff ff ff ff ff   37
7340 : ff cf cf c3 cc cc cf ff   05
7348 : ff cc cc 0c cc cc cf ff   d4
7350 : ff cc cc cc cc cc 3c ff   a6
7358 : ff f0 fc fc fc fc 3c ff   57
7360 : ff 33 f3 f3 f3 f3 f3 ff   23
```

Listing 4. Die Menüzeile für das Fractal-Programm



Soft-Sc

mit allen Raffinessen

Das Scrollen des Bildschirms läßt sich auf dem C64 sehr effektvoll gestalten. Wir stellen Ihnen eine Version dieser Technik vor, die sogar beschleunigendes Scrolling in alle Richtungen erlaubt. Ein hervorragender Effekt, den Sie für eigene Programme übernehmen können.

Auch wenn die neuen 16-Bit-Computer, wie der Amiga oder der Atari ST, den C64 in bezug auf die Grafikfähigkeiten übertreffen, kann er mit ihnen in einem Bereich konkurrieren: die bewegte Grafik. Sie läßt sich auf dem C64 leicht und unkompliziert erzeugen, was auf zwei primäre Ursachen zurückzuführen ist: Zum einen verfügt er über einen Textmodus, in dem er nur wenige Daten zur Bilddarstellung verarbeiten muß. Dies geschieht folglich mit einer hohen Arbeitsgeschwindigkeit. Trotzdem sind hier Grafiken durch umdefinierte Zeichensätze möglich. Zum anderen hat der Videochip des C64 einige Besonderheiten wie die Scroll-Register und den leicht zu programmierenden Raster-Interrupt aufzuweisen.

Bestes Beispiel für eine effektvolle Kombination dieser Eigenschaften ist das Verschieben des Bildschirminhaltes, das sogenannte »Scrolling«: Verschiedene Spiele präsentieren es mittlerweile in Perfektion und erzeugen damit realistische Bewegungssequenzen. Doch auch in der physikalischen Simulation kann es unterstützend eingesetzt werden. In diesem Kurs wollen wir Sie mit der Krönung des Scrollens vertraut machen: Wir stellen ihnen ein zweidimensionales Scrollsystem vor, welches eine beschleunigte Bewegung in alle Bildschirmrichtungen ermöglicht. Das Programm dazu wurde so gestaltet, daß eine Erweiterung leicht möglich ist. Der Quellcode im Listing 2 dient dabei als Basis für die Erläuterungen und ermöglicht Ihnen die Anpassung des Programms an individuelle Erfordernisse. Bild 1 zeigt Ihnen das Scroll-Programm in Aktion (daher auch der verschwommene Hintergrund). Einen weiteren – etwas kuriosen – Effekt sehen Sie in Bild 2: Sogar über den I/O-Bereich bei $D000 läßt sich scrollen.

Physikalische Grundlagen

Um der programmtechnischen Realisierung dieses Vorhabens besser folgen zu können, wollen wir zuerst die Bewegung und ihre physikalischen Grundlagen betrachten. Jede Bewegung eines Körpers läßt sich ganz allgemein als dessen Positionsänderung in einem bestimmten Zeitintervall auffassen. Dies geschieht mit einer Momentangeschwindigkeit »V«, welche ihrerseits konstant oder veränderlich ist. Bei konstanter Geschwindigkeit liegt eine gleichmäßige Bewegung vor. Stellen wir uns hier beispielsweise einen Radfahrer vor, dessen Tacho ständig denselben Wert anzeigt. Pro Sekunde legt er immer die gleiche Wegstrecke zurück, wenn die Geschwindigkeit konstant bleibt. Ist die Bewegung dagegen beschleunigt, verändert sich die Geschwindigkeit mit der Zeit. Bei einer als konstant angenommenen Beschleunigung wird das bewegte Objekt kontinuierlich schneller. Ein fallender Tennisball soll uns dies verdeutlichen: Sobald er unsere Hand verlassen hat, erhöht er seine Geschwindigkeit ständig. Er legt also die gleiche Wegstrecke in immer kürzerer Zeit zurück. Diese zwei Formen der Bewegung lassen sich auch in Diagrammen darstellen, welche den grundsätzlichen Unterschied nochmals verdeutlichen. Betrachten sie hierzu bitte Bild 3.



Bild 1. Das Scrollprogramm in Aktion (aus diesem Grund ist der Hintergrund verschwommen)

Eine weitere Tatsache ist für uns ebenfalls interessant: Jede Wegstrecke hat eine für sie charakteristische Richtung. Da Geschwindigkeit und Beschleunigung die Wegstrecke bestimmen, handelt es sich auch bei ihnen um gerichtete Größen.

Der Vektor zeigt die Richtung

Wir können sie folglich als Vektoren auffassen, und da unser Scrollsystem ebene Bewegungen simulieren soll, werden wir uns auf zwei Dimensionen beschränken. Bild 4 zeigt verschiedene Lagen von Vektoren und deren Unterscheidungsmerkmale: Den Neigungswinkel gegenüber der Horizontalen und die Länge. Doch diese Form der Darstellung ist für unsere Pläne nicht geeignet. Deshalb zerle-

rolling

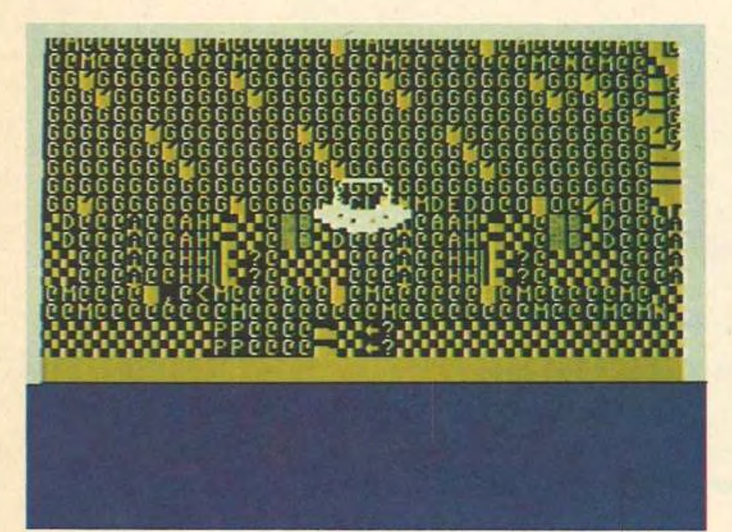


Bild 2. Sogar über den I/O-Bereich läßt sich mit dem »Scrollsystem« scrollen. Ein netter Effekt (diesmal sehr langsam).

gen wir unsere Vektoren wie in Bild 5 in eine horizontale und eine vertikale Komponente, die wir getrennt verarbeiten. Durch deren Kombination ergibt sich dann wieder der ursprüngliche Vektor. Diese Form der Darstellung hat noch einen weiteren Vorteil: Die eindimensionalen Komponenten von Weg, Geschwindigkeit und Beschleunigung dürfen negative Werte annehmen. In diesem Fall ändert sich lediglich die Richtung des Vektors.

Wie verwenden wir diese Erkenntnisse nun auf dem Computer? Wie kann man sie auf den C64 übertragen, und welche Ergebnisse lassen sich erzielen?

Die Beantwortung dieser Fragen soll unser Scrollsystem liefern. Es liegt als Listing 1 im MSE-Format und als Quellcode in Listing 2 vor und zeigt nach dem Programmstart einen grob stilisierten Bergwerksstollen im oberen Teil des Bildschirms. Das Programm liegt im kompaktierten Format mit Basic-Start vor und kann daher mit: LOAD "SCROLL.KOMP",8 geladen werden. Nach dem Start mit RUN wird das eigentliche Scrollprogramm nach $C000, der Hintergrund nach $A000 bis $BFFF verschoben.

In der Mitte des Bildschirms befindet sich ein Raumschiff, das sich mit dem Joystick in Port 2 in alle Richtungen des Bildschirms steuern läßt. Um diesen Effekt zu erzielen, wird der Stollen unter dem Schiff durchgezogen, also gescrollt (Bild 1). Dies geschieht mit einer Geschwindigkeit, die sich bis zu einem Maximalwert steigert. Läßt man den Joystick los, bremst das Raumschiff langsam ab und kommt schließlich zum Stillstand. Auch eine Richtungsumkehr erfolgt nicht abrupt: zuerst wird die Geschwindigkeit bis auf Null reduziert. An diesem Punkt wird die Bewegungsrichtung umgekehrt und erhöht sich wieder. Hier liegt also eine beschleunigte Bewegung vor, sie ist auf dem C64 durchaus effektvoll darzustellen. Doch auf welche Weise wurde dies realisiert? Wie sieht das dazugehörige Programm aus?

Nun, grundsätzlich besteht das Scrollsystem aus Routinen, die im Rasterrinterrupt ablaufen. Dadurch erreichen wir eine flackerfreie Bewegung, deren Berechnung als Hintergrund-Task (im Interrupt) abläuft. Um ein Scrollen des Bildschirms zu erreichen, sind prinzipiell zwei Dinge nötig: Erstens muß das Softscroll-Register im Videochip (bei VIC+17 bzw. VIC+22) laufend verändert werden. Dadurch verschiebt sich der Bildschirm pixelweise, was man als »Softscrollen« bezeichnet. Dies funktioniert allerdings nur bis zu einer Zeichenbreite (also 8 Pixel), dann läuft das VIC-Register über bzw. unter. Deshalb müssen wir zweitens feststellen, wann dies der Fall ist und dann den Bildschirmspeicher um ein Zeichen in die jeweilige Richtung verschieben. Diesen Vorgang bezeichnet man auch als »Hardscrollen«. Unser Scrollen erfolgt also durch das Zusammenspiel von Pixel- und Zeichenbewegung. Außerdem läuft es im Textmodus ab, denn hier benötigt ein Bildschirm nur 1 KByte Speicher und ist damit schnell genug zu verschieben.

Unterscheidung von Geschwindigkeit und Beschleunigung

Bei den folgenden Ausführungen sollten Sie den Quellcode des Scrollsystems, Listing 2, stets im Auge behalten. Der Quellcode ist im Startool-Format geschrieben, ist jedoch leicht an den Hypra-Ass oder unseren neuen Assembler Giga-Ass aus diesem Sonderheft anzupassen. Da der Code relativ lang ist, liefert Bild 6 eine Grobgliederung davon. Zudem erklärt Tabelle 1 die für unser Programm wichtigen Register im Videochip.

Das Listing beginnt mit der Definition oft verwendeter Register und Routinen als Label. Der erste Teil des Programms initialisiert dann einen zweigeteilten Rasterinterrupt und schafft so die Voraussetzungen für die anderen Routinen, welche entweder im oberen oder unteren Bildschirmfenster ablaufen.

Wir wollen uns zuerst mit der pixelweisen Bewegung im oberen Teil beschäftigen, da hier die anfangs gewonnenen, physikalischen Erkenntnisse von entscheidender Bedeutung sind. Der entsprechende Programmabschnitt beginnt

Kurzinfo: Soft-Scrolling

Programmart: Tool zur Scroll-Programmierung
Laden: LOAD "SCROLL.KOMP",8
Start: Nach dem Laden RUN eingeben
Besonderheiten: Das Demo-Programm wird mit dem Joystick in Port 2 gesteuert. Anhand des dokumentierten Quellcodes (StarTool-Format) läßt sich das Programm einfach verändern (Hinweise im Text).
Programmautor: Axel Pretzsch

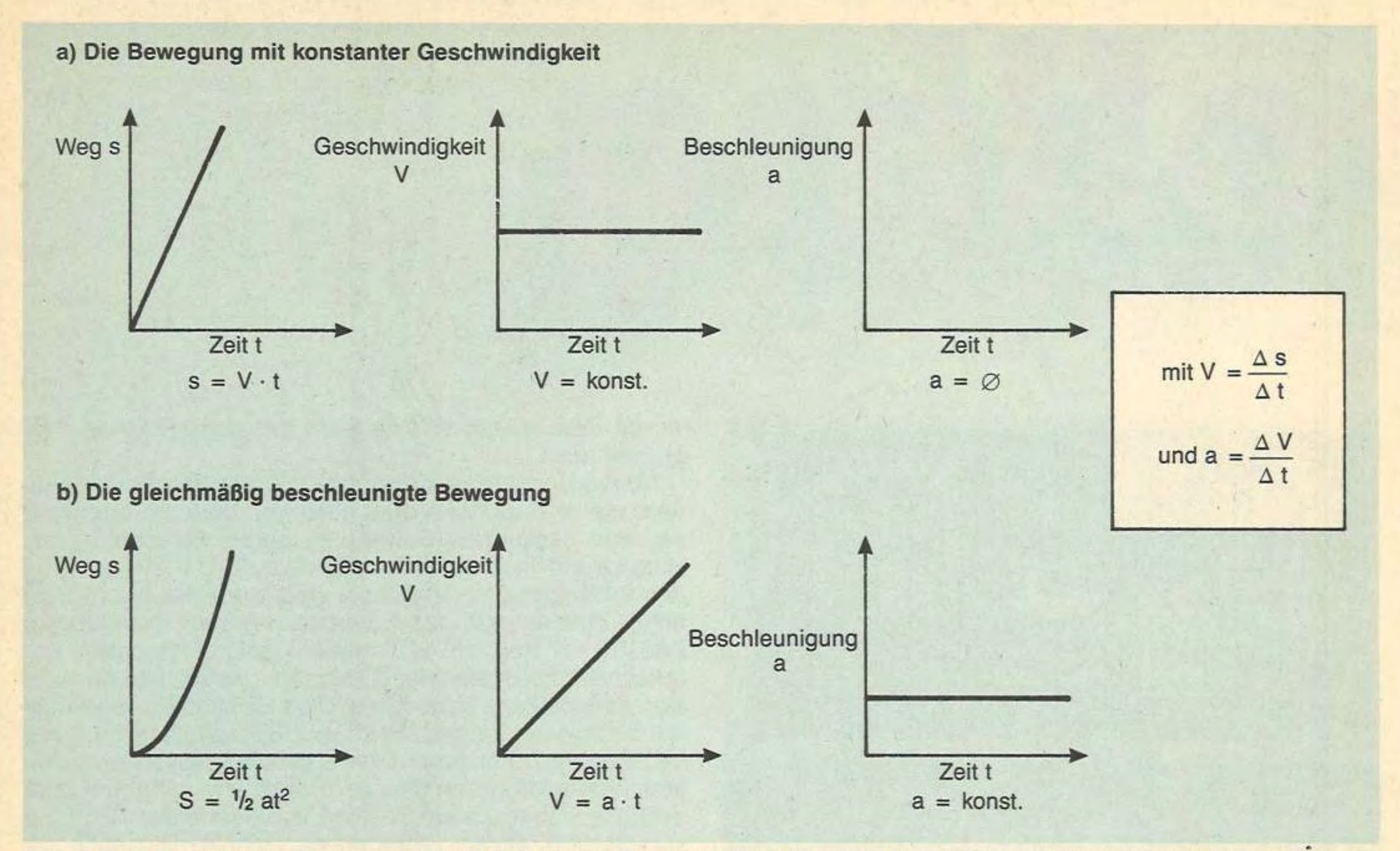


Bild 3. Zwei Formen der Bewegung: a) mit konstanter Geschwindigkeit und b) die gleichmäßig beschleunigte Bewegung

im Quellcode bei Zeile 174 und endet mit Zeile 513. In diesen Routinen werden die beiden Softscroll-Register im Videochip verändert. Dadurch verschiebt sich der Bildschirm pixelweise, was in eine bestimmte Richtung und somit vektoriell geschieht. Also können wir diese Bewegung computergerecht in eine vertikale und eine horizontale Komponente zerlegen, das heißt jedes Softscrollregister wird in einem separaten Programmteil bearbeitet.

Hier kommen nun die von der Zeit abhängigen Funktionen des Weges, der Geschwindigkeit und der Beschleunigung ins Spiel. Da der Rasterinterrupt regelmäßig stattfindet und somit vom einen bis zum nächsten Aufruf unserer Routine immer die gleiche Zeit vergeht, bleibt diese konstant. Wir brauchen also nur die Komponenten des Weges zu verändern, um eine Bewegung zu erzeugen. Dazu addieren wir zum aktuellen Weg die Geschwindigkeit und legen die Summe als neuen Weg fest. Die Geschwindigkeit

Speicherstelle	Bedeutung
VIC=$D000	Basisadresse Videochip
VIC+26:	Legt beim Beschreiben den VIC-IRQ fest Bit 7: VIC-IRQ ein Bit 3: Lightpen-IRQ Bit 2: Sprite/Sprite-Kollisions-IRQ Bit 1: Sprite/Hintergrund-Kollisions-IRQ Bit 0: rasterinterrupt
VIC+25:	Durch Lesen dieses Registers kann ein aufgetretener VIC-IRQ bestimmt werden, Belegung wie VIC+26
VIC+18 und VIC+17 Bit 7:	Zeile für den Rasterinterrupt im MSB/LSB Format
VIC+17:	Bit 0–2 Pixelweises Scrollen, Vertikale Komponente
VIC+22:	Bit 0–2 Horizontale Komponente

Tabelle 1. Fürs Scrollen wichtige Register im VIC

wird wiederum durch die Beschleunigung verändert, welche bei einer Joystickbewegung einsetzt. Außerdem bringt die ständig mit einberechnete (theoretische) Reibung die Scrollbewegung nach dem Loslassen des Joysticks mit der Zeit zum Stehen.

Durch Addieren schneller werden

Die Erledigung dieser Aufgaben übernehmen im Programm einfache Additions- bzw. Subtraktionsroutinen. Da wir in alle Richtungen des Bildschirms scrollen, kann jede Komponente positive und negative Werte annehmen. Die mit der Bewegung in Beziehung stehenden Variablen und Parameter wie Weg, Geschwindigkeit und Beschleunigung sind deshalb als vorzeichenbehaftete 8-Bit-Zahlen angelegt und befinden sich am Ende des Programms (Zeile 715 ff.). Für die vertikale Bewegung wird das Softscrollregister bei VIC+17 verändert, die Funktionsweise der dazu notwendigen Routinen erläutert Ihnen Bild 7 in Verbindung mit dem Quellcode. Die horizontale Bewegung verläuft analog, nur bezieht sie sich auf das Register VIC+22.

Um ein vollständiges Scrollen des Bildschirms zu erzielen, reichen die bis jetzt erörterten Routinen allerdings noch nicht aus, wir benötigen weiterhin Programmteile zum zeichenweisen Verschieben des Bildschirmspeichers. Wir verlassen deshalb den ersten Teil des Interrupts und wenden uns dem zweiten Abschnitt des Scroll-Programms zu,

Ein überdimensionaler Bildschirm

in dem bekanntlich das »Hardscrollen« abläuft. Wie ist dies konzeptiert? Welche Idee steckt dahinter und wie wurde sie realisiert?

Nun, prinzipiell ahmen wir die Bilddarstellung des Computers nach. Dazu organisieren wir den Speicher ähnlich

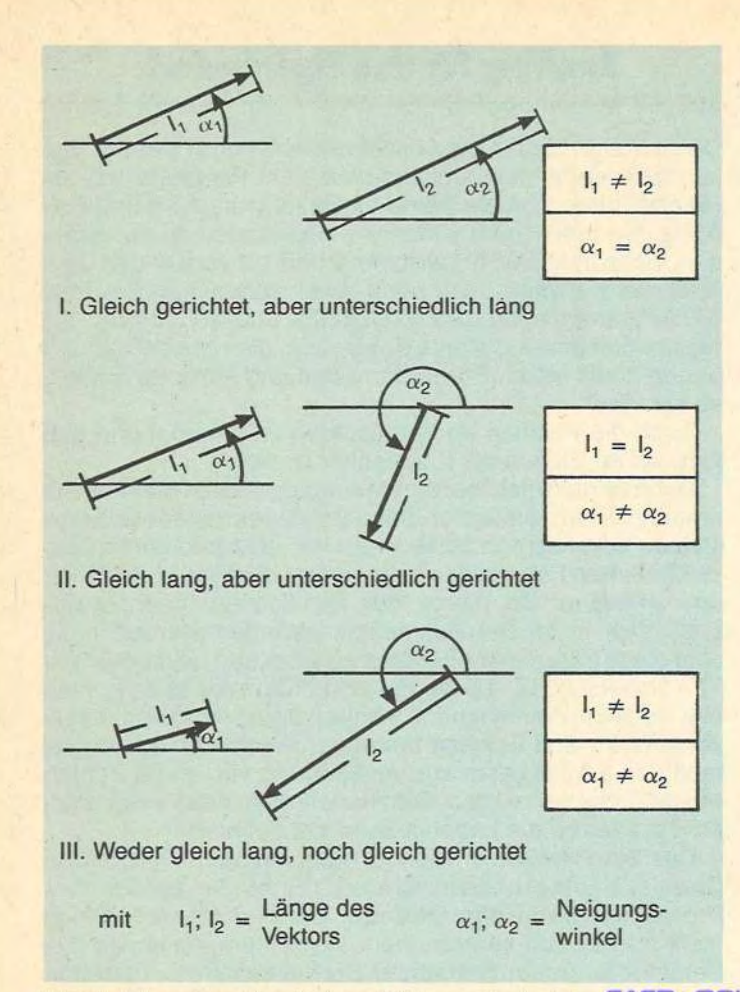


Bild 4. Die unterschiedlichen Vektoren in der Ebene

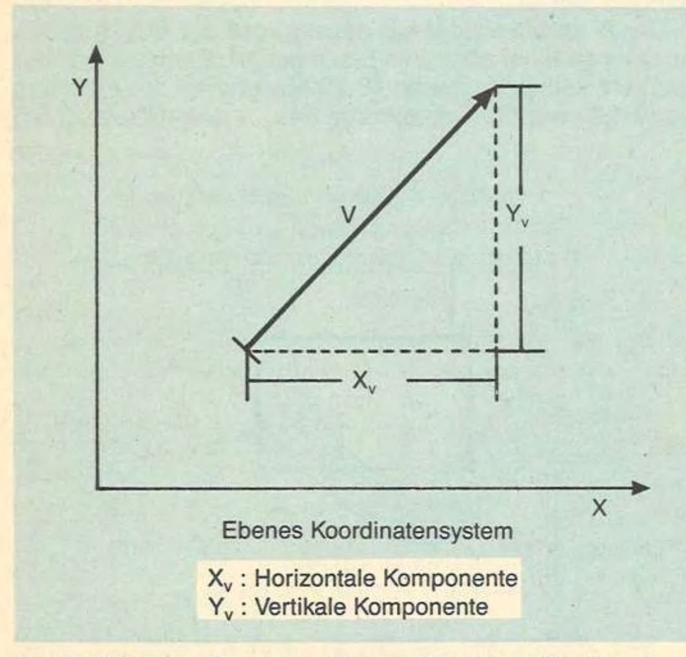


Bild 5. Vektoren, die computergerecht aufbereitetet sind

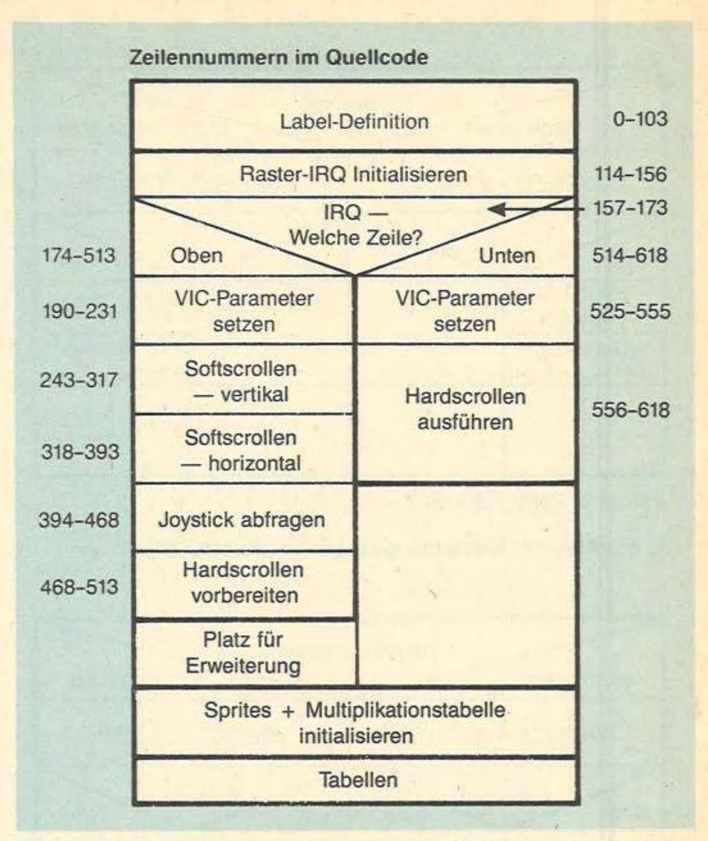


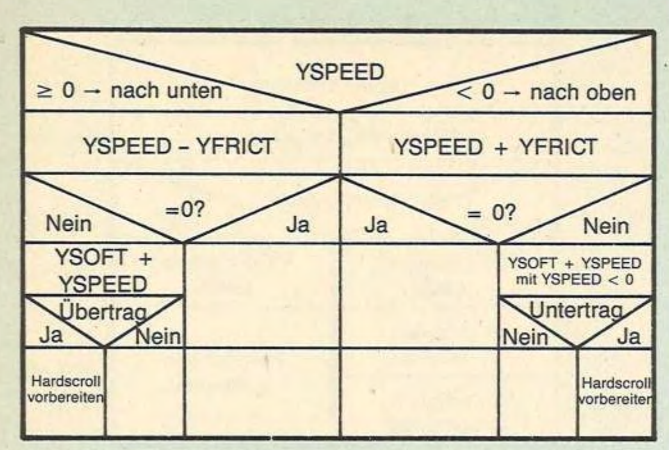
Bild 6. Das Flußdiagramm des Quellcodes veranschaulicht die Funktion des Scrollsystems

wie den Bildschirmspeicher in Zeilen und Spalten, wobei diese Größen wieder in Variablen festgehalten werden. Dieser neu erstellte Bildschirm wird auch als »Scrollscreen« oder »Pseudobildschirm« bezeichnet und ist seinem Vorbild sehr ähnlich, allerdings kann seine Ausdehnung diejenige des »echten« Bildschirms bei weitem überschreiten. Wir sorgen nämlich durch eine Routine dafür, daß auf dem echten Bildschirm immer nur ein Ausschnitt des Scrollscreens zu sehen ist. Diese Aufgabe übernimmt in unserem Scrollsystem die Schleife in den Zeilen 561 bis 618. Welcher Ausschnitt abgebildet wird, hängt von zwei weiteren Variablen ab, in denen die aktuellen X- und Y-Koordinaten gespeichert sind. Sie heißen »XPOINTER« und »YPOINTER«. Werden diese verändert, verschiebt sich logischerweise der Teil im Scrollscreen, welcher abgebildet wird. Und genau dies wird im oberen Teil des Interrupts erledigt, und zwar immer dann, wenn ein Softscrollregister »überläuft«. Somit erreichen wir durch die Kombination von pixel- und zeichenweisem Scrollen ein kontinuierliches Verschieben des Bildschirms. Bild 8 zeigt Ihnen nochmals das Prinzip des Hardscrollens anhand des Scrollscreens.

Veränderung ist angesagt

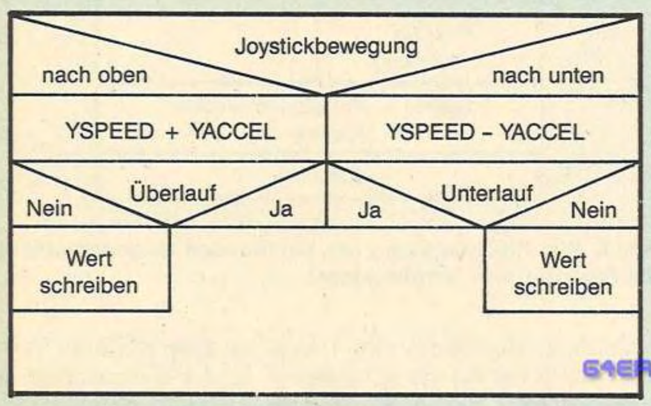
Ein weiterer Hauptbestandteil des Scrollsystems ist die Modifikation der Schleife, die den Abschnitt aus dem Pseudobildschirm holt und darstellt. Dies erfolgt wieder im oberen Teil des Interrupts (bei »HARDPREP«, Zeile 483) und erledigt eine entscheidende Aufgabe: Sie berechnet aus den Zeigern und der Anfangsadresse des Scrollscreens die Speicherstellen und schreibt sie an die entsprechenden Positionen in der im unteren Teil des IRQs ablaufenden Hardscroll-Routine. Somit ist es letztendlich diese Modifikationsroutine, die ein Scrollen ermöglicht. Ohne sie würde die Hardscrollroutine ihre Werte immer aus denselben Speicherplätzen holen und so eben nicht den Eindruck der Bewegung vermitteln. Vielmehr würde der Bildschirm ständig auf sich selbst abgebildet.

Im Rest des Scrollsystems werden die einzelnen Routinen initialisiert, das Ufo-Sprite erzeugt und die Multiplikationstabelle für die Zeile im Scrollscreen berechnet. Das Programm ermöglicht also, zusammenfassend gesagt, eine »saubere« Scrollbewegung, die genau den Gesetzen der Physik gehorcht. Wie übernehmen wir diese nun? Welche Ergebnisse sind damit zu erzielen und auf was muß man bei der Handhabung achten?



Ablauf in Zeile 248 - 277

a) Scrollen mit Momentangeschwindigkeit YSPEED



Ablauf in Zeile 399 - 444

b) Beschleunigen per Joystick

Bedeutung der Parameter		
YSOFT	=	Position
YSPEED	=	Momentangeschwindigkeit
YACCEL	=	Beschleunigung
YFRICT	=	Reibung

Der Wert in YSOFT wird dann bei Zeile 202-211 durch 32 geteilt und ins vertikale Softscrollregister bei VIC+17 geschrieben, was die eigentliche Bewegung verursacht.

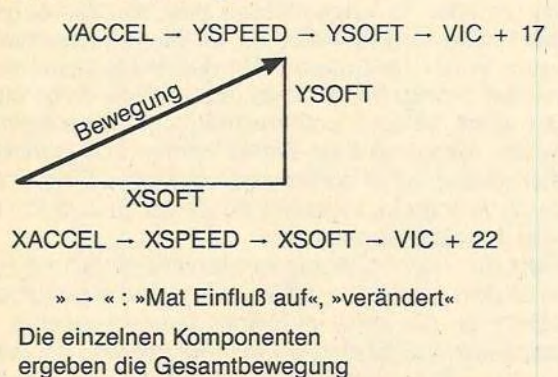


Bild 7. Programmierung der Bewegung in vertikaler Richtung

Scrolling für den Eigenbedarf

Diese Fragen sollen als Abschluß beantwortet werden. Bekanntlich endet der Quellcode mit einer Parametertabelle. Hier befinden sich die Werte für Beschleunigung und Reibung. Sie sind durch einfaches Überschreiben zu verändern, wobei man Werte zwischen 0 und 127 verwenden darf. Außerdem stehen hier noch zwei vorzeichenbehaftete 8-Bit-Parameter namens »XCONST« und »YCONST«. Sie bestimmen eine konstante Bewegung, die von der Reibung unbeeinflußt bleibt. So werden Wind und ähnliche Einflüsse simuliert.

Auch die Position im Scrollscreen ist variabel und legt fest, wo im Stollen Ihr Raumschiff losfliegt.

Doch es muß nicht bei einem simplen Stollen bleiben: Mit umdefiniertem Multicolor-Zeichensatz und einem selbsterstellten Scrollscreen lassen sich bei entsprechenden gestalterischen Fähigkeiten sehr schöne Grafiken à la »Uridium« erstellen. Da hierzu der Scrollscreen höchstwahrscheinlich in seinen Ausmaßen verändert werden muß, sind diese Parameter ebenfalls variabel und heißen »BREITE« und »HOEHE« (Zeile 754 und 755). Weil es sich auch hier um 8-Bit-Werte handelt, können die Abmessungen maximal 255 x 255 Zeichen betragen. Reicht dies nicht aus, muß das Scrollsystem zur Verwendung von 16-Bit-Zahlen umgeschrieben werden. Die Startadresse des Pseudobildschirms wurde als Label in Zeile 112 definiert.

Das Scrollsystem an sich ist natürlich auch erweiterbar. Dies geschieht grundsätzlich durch Einbinden zusätzlicher Programmteile in den Interrupt. Diese dürfen allerdings nicht zu viel Zeit verbrauchen, sie sollten höchstens 1/100 Sekunde lang sein. Betrachten Sie hierzu bitte auch die bekannte Grobgliederung aus Bild 6 und die Speicherbelegung in Bild 9.

Unser Scrollkonzept hat gezeigt, daß der C64 grafisch noch lange nicht zum alten Eisen gehört. Denn bis jetzt beherrscht keine der neuen 16-Bit-Maschinen das Scrolling so perfekt wie der Commodore 64. (Axel Pretzsch/ef)

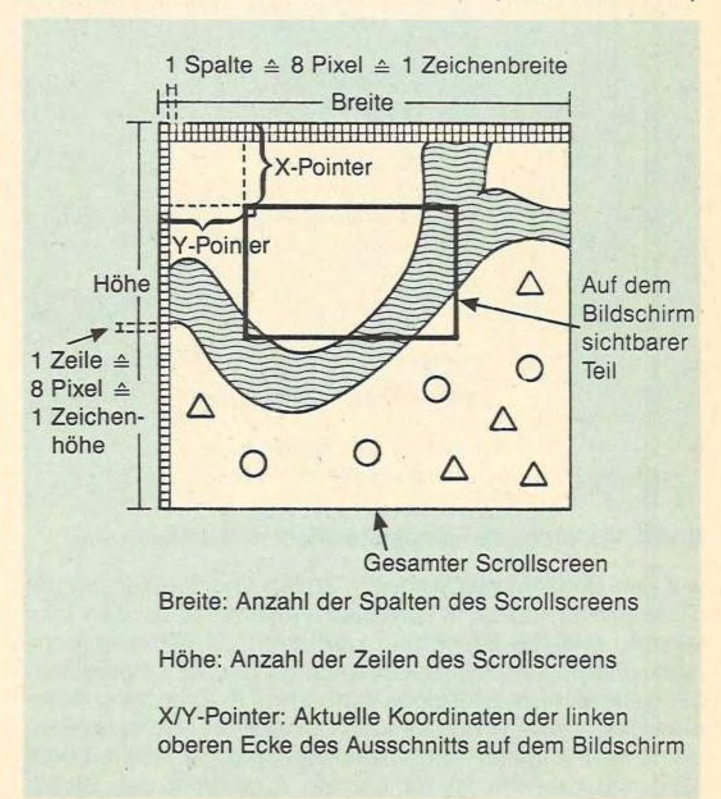


Bild 8. Das Prinzip des Hardscrollens anhand des Scrollscreens im Pseudobildschirm

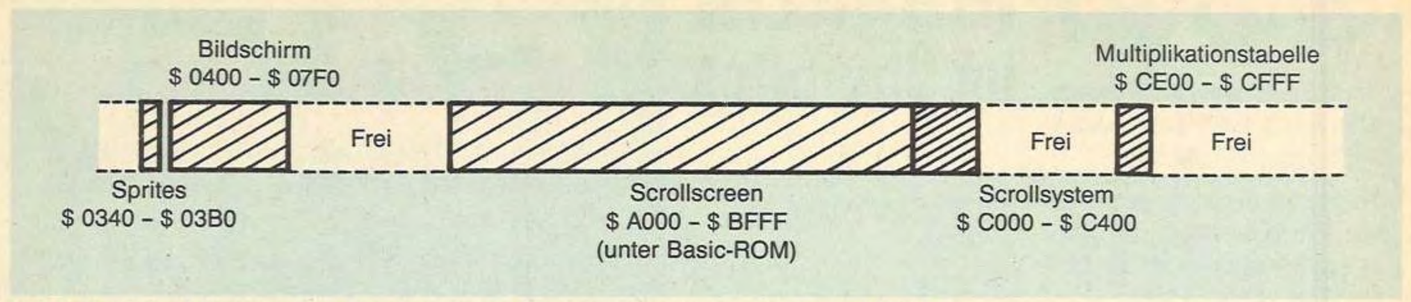


Bild 9. Speicherbelegung des Scrollsystems. Das Hauptprogramm liegt bei $C000.

```
Name : scroll.komp           0801 1034
--------------------------------------
0801 : 0e 08 ca a8 9e 32 30 36  84
0809 : 35 20 46 43 43 00 00 00  7d
0811 : a0 00 b9 69 07 99 00 cd  26
0819 : b9 69 08 99 00 ce b9 69  ec
0821 : 09 99 00 cf c8 d0 eb 4c  4c
0829 : c2 cd 78 a0 ff 84 fb a9  6b
0831 : c6 85 fc a9 36 85 01 8d  dd
0839 : 20 d0 c8 a5 2d d0 02 c6  97
0841 : 2e c6 2d a6 2e e0 0a d0  a6
0849 : 04 c9 16 f0 0f b1 2d 91  2c
0851 : fb a5 fb d0 02 c6 fc c6  10
0859 : fb 4c d3 cd a2 08 a9 01  3c
0861 : 86 2e 85 2d 84 ff 20 50  6f
0869 : ce c9 f3 d0 27 20 50 ce  85
0871 : aa 86 fa c9 04 b0 04 a9  7f
0879 : f3 d0 03 20 50 ce a0 00  97
0881 : 91 2d c8 c6 fa d0 f9 98  03
0889 : 18 65 2d 85 2d 90 02 e6  7d
0891 : 2e 4c 34 ce a0 00 91 2d  77
0899 : e6 2d f0 f3 a9 7c a2 38  4a
08a1 : e4 2e d0 c2 c5 2d d0 be  af
08a9 : a9 37 85 01 a9 fe 8d 20  78
08b1 : d0 58 20 59 a6 4c ae a7  b7
08b9 : a2 ff 86 f7 86 f8 e8 a9  22
08c1 : 01 85 fe a9 7f 85 fd c6  23
08c9 : ff 10 10 e6 fb d0 02 e6  cd
08d1 : fc a9 07 85 ff a0 00 b1  7d
08d9 : fb 85 f9 06 f9 b0 0a a4  6d
08e1 : fe a5 fd 39 f7 00 99 f7  2e
08e9 : 00 8a 0a a8 a5 f7 38 f9  b5
08f1 : e2 ce a5 f8 f9 e3 ce 90  de
08f9 : 0e e0 0b f0 0a e8 38 66  ee
0901 : fd b0 c4 c6 fe f0 bc 8a  e0
0909 : f0 0f a5 f7 38 f9 e0 ce  5e
0911 : 85 f7 a5 f8 f9 e1 ce 85  0f
0919 : f8 a4 fe f0 07 a5 f8 85  ce
0921 : f7 88 84 f8 a5 fd 4a 90  31
0929 : 07 46 f8 66 f7 4c be ce  d9
0931 : bd d2 ce 65 f7 a8 b9 00  63
0939 : cf 60 00 00 00 02 04 07  67
0941 : 11 24 3d 61 8e ab ad ad  38
0949 : ad ad 00 00 00 00 00 40  4e
0951 : 00 60 00 78 00 a0 00 c6  23
0959 : 00 df 00 f1 40 fc e0 ff  f6
0961 : 00 00 00 00 00 00 00 00  62
0969 : f3 a0 20 00 c3 80 ff 8d  10
0971 : ad 08 a9 d0 1d 60 18 07  ea
0979 : 99 b9 05 06 15 04 10 0e  bf
0981 : 01 0f 09 0d 1a a8 03 11  03
0989 : 19 6d 4c 85 90 a2 0b 12  8b
0991 : 95 9d aa ab c1 a6 0c 30  f7
0999 : a5 38 b0 0a 14 28 29 91  12
09a1 : a3 c0 ca 16 21 23 6a a4  38
09a9 : c2 02 17 1c 1e 27 3f 40  4e
09b1 : 66 84 a7 ce cf 13 1f 22  73
09b9 : 2c 2d 2e 31 36 48 50 58  c5
09c1 : 70 bf dc ed f8 1b 24 2f  5d
09c9 : 32 37 3d 4a 68 78 7f 9b  af
09d1 : ac bd ea f0 fa 25 26 2a  fa
09d9 : 2b 33 34 35 39 3a 3b 3c  1c
09e1 : 3e 41 57 59 5a 5b 5f 72  a4
09e9 : 7b 81 86 98 9a 9e 9f ae  54
09f1 : c8 cd df e0 ec ee f7 fc  d4
09f9 : fd 45 56 63 69 6b 6e 71  29
0a01 : 7c 82 88 8a 96 97 9c a1  0e
0a09 : b3 b6 b8 bc d8 dd e2 e6  b3
0a11 : e8 f5 f9 fb fe c8 83 3a  aa
0a19 : 6f a3 c3 a3 d5 ea 2a ac  76
0a21 : 3d 13 06 2f 87 95 24 3f  84
0a29 : b0 e9 e2 f7 f9 5f 04 2a  84
0a31 : 7a f8 bb 1c af 79 3b 5b  04
0a39 : 58 75 39 eb 0a e2 f8 61  76
0a41 : 4c 52 42 b8 bd fc 5d 88  a8
0a49 : 7a f8 be 08 62 c5 ef 27  52
0a51 : 6b 6b 0a ed ab 95 ca e5  10
0a59 : fc 8d 72 b9 5c ae 57 2b  df
0a61 : 95 cb ff 55 ca e5 72 b9  9f
0a69 : 5c ae 57 2b 95 ca e5 72  84
0a71 : b9 5c 8e 1c 02 1c 43 88  9f
0a79 : 71 0e 21 c4 38 8b 68 39  c6
0a81 : 24 5b 41 c9 22 da 0e 49  a0
0a89 : 16 d0 72 48 b6 83 92 45  09
0a91 : b4 14 48 d5 41 ec 91 90  0f
0a99 : 9c d4 44 e6 a2 27 34 52  66
0aa1 : 41 50 8d 04 1d f2 29 a0  3e
0aa9 : 8c 89 90 70 c8 d0 41 be  c2
0ab1 : 23 75 04 64 4c 83 74 8a  e4
0ab9 : c8 37 88 e1 a0 8c 89 90  31
0ac1 : 53 22 c2 04 22 54 11 91  83
0ac9 : 32 0a 64 55 41 c5 22 fa  85
0ad1 : 08 c8 99 06 b1 17 90 58  2b
0ad9 : 22 04 13 11 7d 04 64 4c  06
0ae1 : 83 58 88 10 88 10 5a 23  6d
0ae9 : 21 05 12 2d a0 8c 89 90  6d
0af1 : 51 22 ca 0a 24 59 42 44  e6
0af9 : 08 32 08 b4 82 a1 10 21  6b
0b01 : 17 90 46 44 c8 49 1a 08  ca
0b09 : 2a 11 02 12 44 08 32 08  dc
0b11 : ac 84 8b 28 48 aa 82 32  2f
0b19 : 26 42 48 b4 84 8b 28 49  e1
0b21 : 10 20 d5 22 fa 12 45 54  f9
0b29 : 11 91 32 12 2c 21 10 20  1e
0b31 : a2 44 08 2c 91 7d 09 22  ea
0b39 : aa 08 c8 99 09 12 a0 a2  36
0b41 : 44 08 20 22 fa 0a 24 58  17
0b49 : 41 19 13 21 12 a0 a4 44  41
0b51 : 08 20 22 04 24 56 41 48  fd
0b59 : 8b 08 23 22 64 22 04 22  a1
0b61 : c2 09 88 81 05 22 2c a1  4f
0b69 : 24 5a 41 48 8b 08 23 22  de
0b71 : aa 12 2d 20 80 88 10 52  a5
0b79 : 22 04 14 48 c8 41 01 15  70
0b81 : 90 46 45 84 14 48 d5 41  74
0b89 : 64 88 10 52 22 04 12 91  2d
0b91 : 59 04 64 58 41 7c 88 10  4b
0b99 : 54 22 04 17 c8 ac 82 32  43
0ba1 : 2b 20 b6 44 08 2a 11 02  2d
0ba9 : 0b e4 56 41 19 15 90 5b  17
0bb1 : 22 04 15 08 81 05 f2 2b  7e
0bb9 : 20 8c 8b 48 38 84 40 84  bd
0bc1 : 88 10 5b 22 b2 08 c8 b4  64
0bc9 : 83 88 44 08 48 81 05 b2  ad
0bd1 : 2b 20 8c 8b 48 38 84 40  7a
0bd9 : 84 59 41 6c 8a c8 23 23  a9
0be1 : 41 07 14 8a a8 2f 11 2a  99
0be9 : 08 c8 d0 41 c5 22 aa 0b  e0
0bf1 : c4 4a 82 32 32 10 73 c8  c4
0bf9 : b0 83 24 89 50 46 46 42  7a
0c01 : 04 22 b2 0c 92 25 41 19  ce
0c09 : 1a a8 37 88 b4 83 9c 44  b9
0c11 : a8 23 23 55 07 40 8d 04  6f
0c19 : 1c e2 25 41 19 16 50 6f  7a
0c21 : 88 c8 41 ac 44 a8 3c 62  32
0c29 : 2c a0 a6 44 a8 31 10 e2  f2
0c31 : 1c 43 88 71 0e 21 c4 38  ad
0c39 : b7 c8 7e d7 6f 63 c9 b3  8f
0c41 : 5d eb 5d b7 fe 76 3d 93  a2
0c49 : 13 f7 a2 d8 f2 5e bd 14  5d
0c51 : 3f 23 d6 e2 87 c5 7b 5a  7d
0c59 : ec 3b 6f 53 ba 9f d7 82  36
0c61 : 9d de 0c 2a f3 b8 ef e4  44
0c69 : c5 f6 79 4c 8b bb d8 91  ae
0c71 : fb f1 3d 7e 27 f2 62 fb  0f
0c79 : 3c a6 6a 70 b1 ae e1 fb  c1
0c81 : f3 93 f5 6d a6 7e da 7f  32
0c89 : ad b4 d0 ce 4e b6 72 7c  fc
0c91 : 59 c9 da ce 7f ca 63 d9  ef
0c99 : 31 61 ea 79 af ff 06 3c  f0
0ca1 : 91 3f fc 58 f6 f4 50 ab  cc
0ca9 : d7 a2 7f 35 82 8f 8d b1  96
0cb1 : 37 9d 75 eb d1 3f 98 c1  8f
0cb9 : 47 c6 d8 9a cc 4f 6e c4  77
0cc1 : fd e8 ac d5 7a f4 52 3f  2f
0cc9 : 0b 34 63 c3 f5 6d bf 0b  1f
0cd1 : 34 fb 98 ec d4 af 23 d0  bf
0cd9 : b1 fc d6 4b 82 16 3d 9a  2a
0ce1 : cd 4f 72 47 e1 64 b8 31  62
0ce9 : d9 8d 5e 47 a1 63 f9 ac  80
0cf1 : 97 04 2c 7b 35 98 db e9  60
0cf9 : 1f e0 33 46 ae 1f 37 6d  ba
0d01 : fc d6 4b 83 80 c7 b3 59  73
0d09 : a9 93 23 f9 ac 97 07 01  29
0d11 : 8f 66 b3 1a bc 9f 86 7b  dd
0d19 : 24 e6 67 b2 47 ee 33 46  26
0d21 : 3c 3a d9 cf dc 66 9f 73  51
0d29 : 1d 9a 95 e4 7a e3 1f cc  72
0d31 : 64 b8 2e 31 ec c6 6a 7b  49
0d39 : 92 3f 71 92 e0 fe 6c c6  5f
0d41 : af 23 d7 18 fe 63 25 c1  9e
0d49 : 71 8f 66 33 1b 7d 23 fb  a4
0d51 : 4c d1 ab 87 af 9c fe 63  04
0d59 : 25 c1 b4 c7 b3 19 a9 93  57
0d61 : 23 f9 8c 97 06 d3 1e cc  a8
0d69 : 66 35 79 3f 0d 26 49 cc  71
0d71 : d2 64 8f af 6f 7f 8d 63  3f
0d79 : 7f 8d de 7d 7b 7b fd fe  b0
```

Listing 1. »SCROLL.KOMP«, Multidirektionales Scrolling, per Joysick gesteuert. Start mit RUN. Bitte mit dem MSE (siehe Seite 159) eingeben.

64'er Magazin im Überblick

Diese 64'er- Ausgaben bekommen Sie noch bei Markt & Technik für jeweils 6,50 DM.

Tragen Sie die Nummer der gewünschten Ausgabe (z.B. 3/88) in den Bestellabschnitt der Zahlkarte nach Seite 34 ein.

12/86: Übersicht: Hardware- Erweiterungen Bauanleitung: Centronics- Interface Listing des Monats: Floppy-Speeder "Exos V3"

1/87: Spiele: Die Renner ´86, Billigspiele im Test Farbmonitore im Vergleich / Großer Einsteigerteil: So fängt man an

3/87: Zum Abtippen: Kopierprogramm der Spitzenklasse / Disketten: Markenqualität gegen No-Name- Produkte C128 : Speichererweiterungen im Test

4/87: Programmiersprachen: So arbeiten Profis Listing des Monats: Terminalprogramm "Proterm V6" Test: Farbfernsehgeräte als Monitorersatz

5/87: Fractals: Die Welt der Apfelmännchen Kaufhilfe: Die besten Floppy- Speeder 3½-Zoll- Floppy für den C64

2/88: Desktop Publishing live: Zeitung machen mit dem C64 / Tolles Malprogramm zum Abtippen

3/88: Brennpunkt Spiele: Spiele per Telefon u. a. Kopierprogramme im Vergleich

4/88: Gibt es einen neuen C64 ? / Alles über Btx und Datenfernübertragung / Große Checkliste zum Kauf von Software

5/88: C 64 contra Amiga, Atari & Co, Vergleichstest: Drucker / Im Härtetest: neuer Super-Joystick / Großer Einsteiger-Sonderteil

6/88: Keyboards am C64 / Markendisketten im Härtetest / Test: Floppy-Speede Neuer Kurs: Assembler

8/88: Tips und Tricks zu Druckern / Basic-Kurs für Einsteiger / Alles über RAM, ROM, EPROM & Co.

9/88: Neuer Kurs: Drucker professionell nutzen Messen, Steuern, Regeln: Profigeräte im Test / Public Domain- Spiele

10/88: Test: Modems und Akustikkoppler Listing des Monats: Super- Strategie- Spiel Musikhardware im Vergleich

11/88: Publish C64: Professionelles Druckprogramm zum Abtippen / Test: Malprogramm Giga- Paint Ratgeber Druckerkauf

12/88: Weihnachts- Special: Die besten Geschenkideen / Geheimtip: Monitor für 40,-DM / Bauanleitung: Drucker- Interface

1/89: Die besten Druckprogramme / 20 Zeiler zum Abtippen / Malprogramme für den C128 im Vergleich Jahresinhaltsverzeichnis

2/89: Test: Schnellster Basic- Compiler Listing: "Master Copy Plus" / Spiele ´88 Computerschreibtisch zum Spartarif

3/89: Kaufhilfe: Floppies, Drucker, Monitore Bauanleitung: 256 KByte Zusatzspeicher / Software-Test: Geos 2.0 ist da / Viren im C64

4/89 C 64-Longplay: Uridium komplett durchgespielt / Listing des Monats: Think Twice, ein Knobelspiel/ C 64 Extra

5/89 Lohnt sich ein Interface ? / Test: Die besten Mailboxen / Druckerständer für 10 Mark

6/89 Großer Diskettenvergleichstest/ Listings des Monats: Textverarbeitungsprogramme Text II / Spielekurs Teil 1

7/89: Spiele-Extra: Spielesteckbriefe zum Sammeln/ Zeichensätze selbst gemacht/ Test: Joysticks

Mit diesen Sammelboxen sind Ihre Ausgaben immer sortiert und griffbereit.

Eine Sammelbox faßt einen vollständigen Jahrgang mit 12 Ausgaben und kostet 14,– DM.





Sonderhefte im Überblick

Die 64er Sonderhefte bieten Ihnen umfassende Informationen in komprimierter Form zu speziellen Themen rund um die Commodore C 64, C 128, C 16/116, VC 20 und denPlus/4. Diese Ausgaben hat Ihr Händler vorrätig - oder er bestellt sie gerne für Sie.

DRUCKER



SH 9904: GRAFIK & DRUCKER
80- Zeichen- Karte zum Abtippen / Hardcopy- Routinen für viele Drucker



SH 0018: DRUCKER
Listing: professionelle Textverarbeitung für den MPS 801 / Matrixdrucker im Test



SH 0032: FLOPPYLAUFWERKE UND DRUCKER
Tips&Tools / RAM- Erweiterung des C64 / Druckerroutinen

HARDWARE

SH 0013: HARDWARE
Ein- Chip- Microcomputer / Bauanleitungen: MIDI-Interface, Speicheroszilloskop, IC- Tester

FLOPPY, DATASETTE



SH 9905: FLOPPY / DATASETTE
Disketten kopieren mit Hypra-Copy/ 10mal schneller laden mit Turbo Tape de Luxe



SH 0009: FLOPPY / DATEIVERWALTUNG
Floppy- Beschleuniger im Vergleichstest / Arbeiten mit dBase II / C 128- Diskmonitor



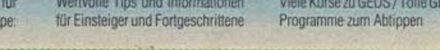
SH 0015: FLOPPY / DATASETTE
Reparaturanleitung: Erste Hilfe für die Diskettenstation / Hypratape: das Super- Turbotape



SH 0025: FLOPPY- LAUFWERKE
Wertvolle Tips und Informationen für Einsteiger und Fortgeschrittene



SH 0028: GEOS / DATEIVERWALTUNG
Viele Kurse zu GEOS / Tolle GEOS-Programme zum Abtippen



GRAFIK, SOUND



SH 0011: GRAFIK, MUSIK, ANWENDUNGEN
50 Seiten Musikprogrammierung / Vielseitige Businessgrafik



SH 0020: GRAFIK
Grafik- Programmierung / Bewegungen



SH 0023: GRAFIK, ANWENDUNGEN
Außergewöhnliche Anwendungen auf dem C 64 zum Abtippen



SH 0027: GRAFIK
AMICA Paint: Malprogramm



SH 0034: GRAFIK, SIMULATION, LERNEN
Konstruieren mit dem C64 / Kurvendiskussion / Einstieg in die Digitaltechnik

C 64 - EINSTIEG



SH 0005: C 64- GRUNDWISSEN
Vom ersten Einschalten bis zum eigenen Programm / Grundlagen, Tips und Tricks



SH 0016: EINSTEIGER 2
Spriteanimation: Zeichentrickfilm mit dem Computer / GEOS, die neue Benutzeroberfläche



SH 0019: EINSTEIGER 3
Basic- Kurs / Programm- Übersicht



SH 0026: RUND UM DEN C 64
Der C 64 verständlich für Alle mit ausführlichen Kursen

C 128



SH 0001: C 128
Das können C 128 und C 128 D / Vergleich: C 128- C 64 / die passende Peripherie



SH 0010: C 128 II
Die Geheimnisse von CP/M / Kompletter C 128- Schaltplan / Grafik für Einsteiger



SH 0022: C 128 III
Farbiges Scrolling im 80- Zeichen-Modus / 8- Sekunden- Kopierprogramm



SH 0029: C 128
Starke Software für C 128/C 128 D/ Alles über den neuen C 128 D im Blechgehäuse



SH 0036: C 128
Power 128: Directory komfortabel organisieren / Haushaltsbuch: Finanzen im Griff / 3D- Landschaften aus dem Computer

C 16/116, VC 20, PLUS/4



SH 0003: C 16/116, VC 20, PLUS/4
Listings für Spiele, Grafik, Tips &Tricks / Anwendungen: Dateiverwaltung, VC 20 mit Musik



SH 0008: PLUS/4 UND C16
Übersicht: Zeropage und wichtige Systemadressen / Grundlagen und viele Listings

SPIELE



SH 9902: ABENTEUER-SPIELE
45 Seiten Adventure-Programmierkurs / Listings und Schritt-für-Schritt-Lösungen



SH 9903: SPIELE
Top- Spiele- Listings für C 64 und VC 20 / Große Spiele- Marktübersicht



SH 0004: ABENTEUER-SPIELE
Kurs: Programmierung von Grafik, Parser und künstlicher Intelligenz / Viele Adventures



SH 0017: SPIELE FÜR C64 UND C 128
So programmiert man Scrolling / Strategiespiele: Grips ist gefragt



SH 0030: SPIELE FÜR C64 UND C 128
Tolle Spiele zum Abtippen für C 64/ C 128 / Spieleprogrammierung

TIPS&TRICKS, ANWENDUNGEN



SH 9901: TIPS&TRICKS
Befehlserweiterungen für Betriebssystem und Floppy / Unentbehrliche Programmierhilfen



SH 9906: AUSGEWÄHLTE SUPERLISTINGS
Die besten Programme aus den 64er- Magazinen 1984/85



SH 9907: ANWENDUNGEN/ DFÜ
Terminal und Mailboxprogramm zum Abtippen / Der C 64 als Winzer



SH 0002: TIPS&TRICKS
Zeichensatz- und Sprite- Editor / Interrupt- Joystickabfrage / 27 nützliche Einzeiler



SH 0024: TIPS, TRICKS & TOOLS
Die besten Peeks und Pokes sowie Utilities mit Pfiff



SH 0031: DFÜ, MUSIK, MESSEN - STEUERN - REGELN
Alles über DFÜ / BTX von A-Z / Grundlagen / Bauanleitungen



SH 0033: TIPS,TRICKS & TOOLS
Basic- Control- System / Titelgenerator / Digitale Super- Sounds / Betriebssysteme im Vergleich

PROGRAMMIER- UND MASCHINENSPRACHE



SH 0007: PEEKS&POKES
"Maschinen- Power" mit Basic / Multitasking: 2 Basic- Programme laufen nebeneinander / Peeks und Pokes zum C 128



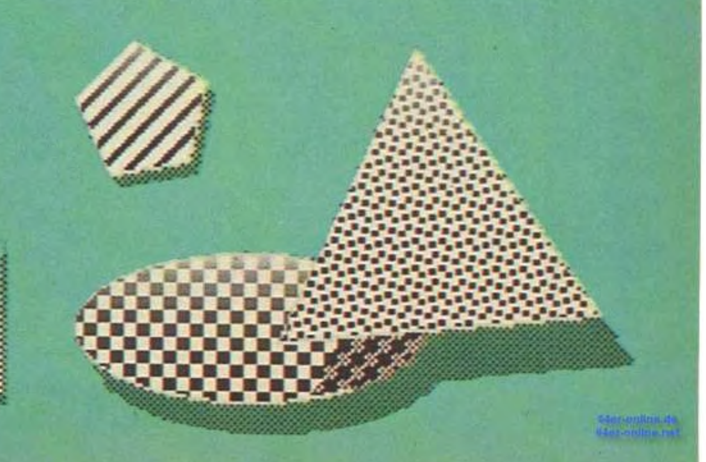
SH 0012: PROGRAMMIERSPRACHEN
Pascal, Comal, Prolog, C und Forth/ Vergleich: Basic- Compiler



SH 0021: ASSEMBLER UND BASIC
Giga- Ass: Hypra- Ass hoch 2 / Paradoxon- Basic: 50000 Basic Bytes free

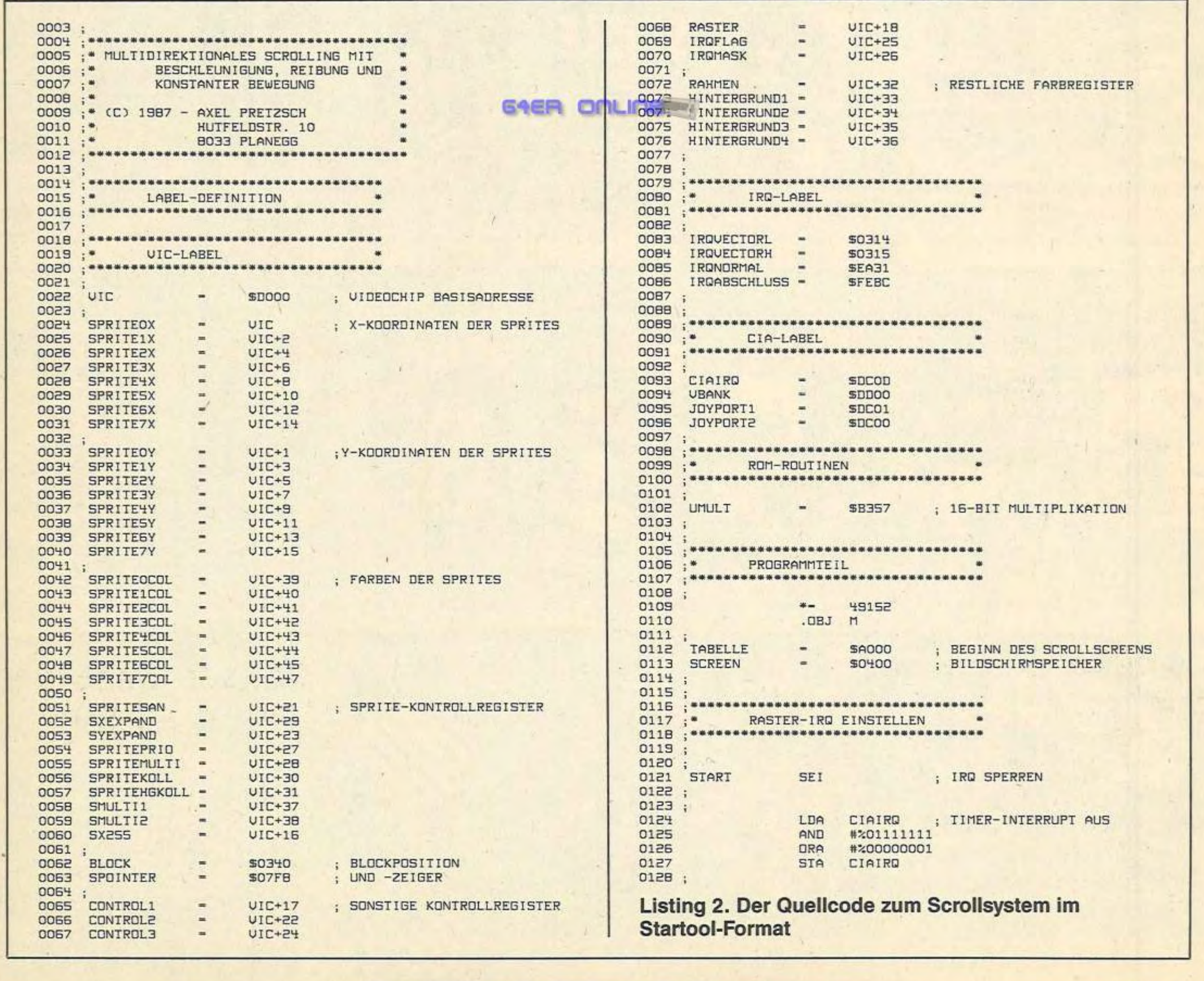


SH 0035: ASSEMBLER
Abgeschlossene Kurse für Anfänger und Fortgeschrittene



```
0d81 : ff 1b a1 bf c6 e8 c8 fc  3f
0d89 : 2c 97 01 67 66 eb d0 b3  1e
0d91 : 0f c7 9c fc 2c d3 ee 16  94
0d99 : 76 6e bd 0b 30 fc 79 cf  88
0da1 : dc 64 b8 3e 96 76 6e bd  f8
0da9 : 71 92 3f 71 9a 7d cf a5  81
0db1 : 9d 9b af 5c 64 9e 5e 7b  3f
0db9 : 31 d5 ff a1 8f 72 d9 2d  57
0dc1 : 05 e6 60 d2 66 5f 4f 83  51
0dc9 : 41 79 bf a0 e5 f8 38 22  16
0dd1 : 97 fc fa f9 9c b6 3f cb  f8
0dd9 : 64 5f 6c 89 ff d9 c9 c6  bd
0de1 : e0 bf f3 b1 ec 98 b0 ff  2a
0de9 : 7f ff 7f e6 60 a3 e6 bc  5d
0df1 : 91 3f 3b 1e de 8b 1e d4  21
0df9 : 31 65 ac 5b 9b 95 7f f4  c2
0e01 : 32 3d 56 28 78 4f 5e 8a  fd
0e09 : 18 9e dd 8a 43 c6 87 ab  19
0e11 : 8a bd 0c ac 79 d6 a5 0a  0b
0e19 : 7b 53 eb d4 a1 95 b5 3f  ef
0e21 : 5e a5 0a 7c 09 fc 8a 94  30
0e29 : 32 b8 13 9a 94 29 e9 4f  a8
0e31 : f5 d4 a1 95 a5 3f 9d 52  1b
0e39 : 85 3c f9 e5 a3 43 2b 3e  95
0e41 : 7e 45 1a 14 f9 73 f8 f4  74
0e49 : 68 65 72 e7 c6 a3 42 9d  cb
0e51 : c9 ff b5 1a 19 57 27 fb  ab
0e59 : a8 d0 a7 0c f0 52 a1 95  28
0e61 : 0c fa 34 a8 53 cc 9f b1  8a
0e69 : 4a 86 56 64 ed a5 42 9e  6b
0e71 : 6c f7 29 50 ca cd 9e 2a  17
0e79 : 54 29 f9 73 f7 69 50 ca  f0
0e81 : f2 e7 49 bf 76 8d dc 3d  73
0e89 : 2e 0c 3e 26 2a f2 3f fc  43
0e91 : 98 f6 93 1f cb 63 d9 ec  86
0e99 : 2b ca d7 2b ca d8 fa b4  4d
0ea1 : 99 f2 fe bc 75 f9 7e 24  74
0ea9 : f7 7f a7 ef eb 7c dd 6c  3a
0eb1 : fd 7f 1f 2d 79 be 3e 5a  17
0eb9 : f3 3f de be 7e bf df a4  8a
0ec1 : c8 bf cc 34 de d2 64 34  a1
0ec9 : 5e cf 64 98 fc 88 5d 97  f4
0ed1 : c8 bb b9 57 cf c7 e4 79  92
0ed9 : 9f 99 99 6d bb b9 57 f1  24
0ee1 : 63 d8 f3 3f 2b 32 d6 2d  8f
0ee9 : ca bf 8b 1e 93 f3 b3 2f  3f
0ef1 : 8d 16 e5 5f 3e 1a ef 74  4c
0ef9 : e2 c7 a5 0d ac be ec db  f6
0f01 : 95 7c f8 6b 3c e9 a1 73  00
0f09 : d4 e2 7a 38 a1 6b dc 38  4d
0f11 : a4 97 67 ad b3 e4 6c dc  de
0f19 : d9 ff 1b 30 6c e9 ec 9f  c8
0f21 : d3 f7 0b 0b 5d cd e1 2f  3e
0f29 : 22 a2 a5 45 4a 8a 94 96  27
0f31 : 4e 3f c9 f8 dc e7 7e 47  46
0f39 : 76 b8 fe 67 e3 b0 ef 39  ae
0f41 : 5c 2b 2d 75 a5 4b a5 13  9e
0f49 : 5d 4e a5 4a f5 d5 69 d1  d7
0f51 : ae 3f 3a 87 28 72 87 28  a3
0f59 : 72 87 28 72 87 28 72 87  7a
0f61 : 28 72 b5 5a ad 56 ab 55  62
0f69 : aa d5 6a b5 5a ad 56 ab  13
0f71 : 55 aa d5 6a b5 5a ad 56  70
0f79 : ab 55 aa d5 6a b5 5a ad  4d
0f81 : 56 ab 55 aa d5 6a b5 5a  94
0f89 : ad 56 ab 55 aa d5 6a b5  65
0f91 : 5a ad 56 ab 55 aa d5 6a  a4
0f99 : b5 5a ad 56 ab 55 aa d5  6d
0fa1 : 6a b5 5a ad 56 ab 55 aa  a0
0fa9 : d5 68 fd ab 5e bf 0b 85  c2
0fb1 : 76 ed 4a 94 68 d2 a5 34  5f
0fb9 : d0 41 66 ce ae ae 46 46  a3
0fc1 : 86 85 ab 55 ab 58 b1 56  90
0fc9 : ad eb d9 39 3c ee 76 b6  8c
0fd1 : b5 3a 7b bb bc 3e 1c b2  8d
0fd9 : df bf 6e df 7b bd c4 e2  ae
0fe1 : 47 1f 17 8b cf e7 a2 6f  95
0fe9 : 6f 74 3a 1b ed f7 7f bf  a0
0ff1 : e8 7a 1e 8f a3 c6 e3 6b  67
0ff9 : eb ec 6c 7a 5e 97 a7 e9  da
1001 : f4 7a 3d 2e 97 4f a7 e0  9c
1009 : f0 68 e8 f5 3a 9e 1f 0f  59
1011 : a9 ea 7a be af ad eb 75  a9
1019 : 7a be 27 89 a7 a7 eb fa  4b
1021 : fe c7 b1 ec fb 3e d7 b5  89
1029 : e2 f8 be df b7 c7 e3 8f  9b
1031 : 25 7e d0 20 4c 41 55 54  9a
```

Listing 1. »SCROLL.KOMP« (Schluß)



```
0003 ;
0004 ;*************************************
0005 ;* MULTIDIREKTIONALES SCROLLING MIT   *
0006 ;*      BESCHLEUNIGUNG, REIBUNG UND   *
0007 ;*      KONSTANTER BEWEGUNG           *
0008 ;*                                    *
0009 ;* (C) 1987 - AXEL PRETZSCH           *
0010 ;*            HUTFELDSTR. 10          *
0011 ;*            8033 PLANEGG            *
0012 ;*************************************
0013 ;
0014 ;**********************************
0015 ;*      LABEL-DEFINITION          *
0016 ;**********************************
0017 ;
0018 ;**********************************
0019 ;*      VIC-LABEL                 *
0020 ;**********************************
0021 ;
0022  VIC          =     $D000     ; VIDEOCHIP BASISADRESSE
0023 ;
0024  SPRITE0X     =     VIC       ; X-KOORDINATEN DER SPRITES
0025  SPRITE1X     =     VIC+2
0026  SPRITE2X     =     VIC+4
0027  SPRITE3X     =     VIC+6
0028  SPRITE4X     =     VIC+8
0029  SPRITE5X     =     VIC+10
0030  SPRITE6X     =     VIC+12
0031  SPRITE7X     =     VIC+14
0032 ;
0033  SPRITE0Y     =     VIC+1     ;Y-KOORDINATEN DER SPRITES
0034  SPRITE1Y     =     VIC+3
0035  SPRITE2Y     =     VIC+5
0036  SPRITE3Y     =     VIC+7
0037  SPRITE4Y     =     VIC+9
0038  SPRITE5Y     =     VIC+11
0039  SPRITE6Y     =     VIC+13
0040  SPRITE7Y     =     VIC+15
0041 ;
0042  SPRITE0COL   =     VIC+39    ; FARBEN DER SPRITES
0043  SPRITE1COL   =     VIC+40
0044  SPRITE2COL   =     VIC+41
0045  SPRITE3COL   =     VIC+42
0046  SPRITE4COL   =     VIC+43
0047  SPRITE5COL   =     VIC+44
0048  SPRITE6COL   =     VIC+45
0049  SPRITE7COL   =     VIC+47
0050 ;
0051  SPRITESAN    =     VIC+21    ; SPRITE-KONTROLLREGISTER
0052  SXEXPAND     =     VIC+29
0053  SYEXPAND     =     VIC+23
0054  SPRITEPRIO   =     VIC+27
0055  SPRITEMULTI  =     VIC+28
0056  SPRITEKOLL   =     VIC+30
0057  SPRITEHGKOLL =     VIC+31
0058  SMULTI1      =     VIC+37
0059  SMULTI2      =     VIC+38
0060  SX255        =     VIC+16
0061 ;
0062  BLOCK        =     $0340     ; BLOCKPOSITION
0063  SPOINTER     =     $07F8     ; UND -ZEIGER
0064 ;
0065  CONTROL1     =     VIC+17    ; SONSTIGE KONTROLLREGISTER
0066  CONTROL2     =     VIC+22
0067  CONTROL3     =     VIC+24
0068  RASTER       =     VIC+18
0069  IRQFLAG      =     VIC+25
0070  IRQMASK      =     VIC+26
0071 ;
0072  RAHMEN       =     VIC+32    ; RESTLICHE FARBREGISTER
0073  HINTERGRUND1 =     VIC+33
0074  HINTERGRUND2 =     VIC+34
0075  HINTERGRUND3 =     VIC+35
0076  HINTERGRUND4 =     VIC+36
0077 ;
0078 ;
0079 ;**********************************
0080 ;*      IRQ-LABEL                 *
0081 ;**********************************
0082 ;
0083  IRQVECTORL   =     $0314
0084  IRQVECTORH   =     $0315
0085  IRQNORMAL    =     $EA31
0086  IRQABSCHLUSS =     $FEBC
0087 ;
0088 ;
0089 ;**********************************
0090 ;*      CIA-LABEL                 *
0091 ;**********************************
0092 ;
0093  CIAIRQ       =     $DC0D
0094  VBANK        =     $DD00
0095  JOYPORT1     =     $DC01
0096  JOYPORT2     =     $DC00
0097 ;
0098 ;**********************************
0099 ;*      ROM-ROUTINEN              *
0100 ;**********************************
0101 ;
0102  UMULT        =     $B357     ; 16-BIT MULTIPLIKATION
0103 ;
0104 ;
0105 ;**********************************
0106 ;*      PROGRAMMTEIL              *
0107 ;**********************************
0108 ;
0109               *=    49152
0110               .OBJ  M
0111 ;
0112  TABELLE      =     $A000     ; BEGINN DES SCROLLSCREENS
0113  SCREEN       =     $0400     ; BILDSCHIRMSPEICHER
0114 ;
0115 ;
0116 ;**********************************
0117 ;*      RASTER-IRQ EINSTELLEN     *
0118 ;**********************************
0119 ;
0120 ;
0121  START        SEI             ; IRQ SPERREN
0122 ;
0123 ;
0124               LDA   CIAIRQ    ; TIMER-INTERRUPT AUS
0125               AND   #%01111111
0126               ORA   #%00000001
0127               STA   CIAIRQ
0128 ;
```

Listing 2. Der Quellcode zum Scrollsystem im Startool-Format

64'er SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt abtippen!

Die aktuelle Diskette zum Heft:

64'er-Sonderheft 45: Faszinierende GRAFIK-Dimensionen

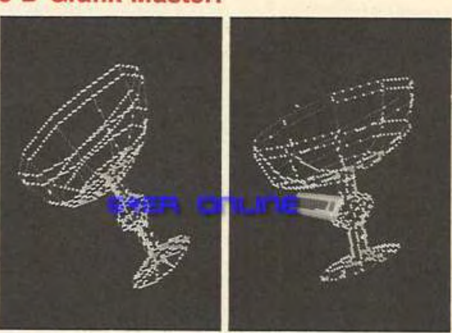
Prograf: Dreidimensionales Zeichnen von HiRes-Grafiken wird mit dieser Befehlserweiterung zum Kinderspiel. Die Ergebnisse können sich auf jedem Epson-kompatiblen Drucker bewundern lassen: denn Prograf nutzt die hohe Auflösung von 640 x 400 Punkten voll aus.

Genaue Programmbeschreibung auf S. 38ff in diesem Heft.

Computerlandschaften: Werden Sie mit Ihrem Computer zum Landschaftsgestalter. Mit zufallsgesteuerten Fractals entstehen in wenigen Schritten eindrucksvolle Landschaften mit Hügeln, Tälern und Seen.

Genaue Programmbeschreibung auf S. 14ff in diesem Heft.

3-D-Grafik-Master:



Dreidimensionale, plastische Körper aus jedem beliebigen Blickwinkel betrachten und sogar um jede mögliche Achse drehen – das ermöglicht Ihnen dieses Grafik-Animationsprogramm. Wenn Sie wollen, auch in Echtzeit.

Genaue Programmbeschreibung auf S. 73ff in diesem Heft.

Amica-Paint-Erweiterungen: Das tolle Malprogramm »Amica-Paint« läßt sich noch weiter ausbauen. Neue Maustreiber für die 1351- und die NCE-Maus, ein verbesserter Schnellader, neue Dia-Show, zwei zusätzliche Erweiterungen lassen das Herz eines jeden Amica-Paint-Anwenders höher schlagen.

Genaue Programmbeschreibung auf S. 54ff in diesem Heft.

Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des 64'er-Sonderheftes 45 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 15945

DM 29,90* sFr 24,90*/öS 299,-*

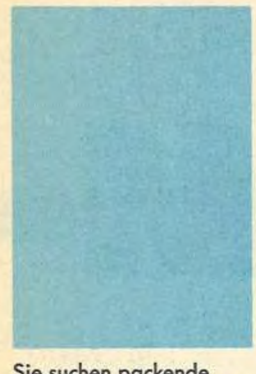


10 Leerdisketten 5¼"
zum Sonderpreis von DM 19,90
Bestell-Nr. 39000, 2seitig,
doppelte Dichte DS/DD, 40 Spuren,
48 tpi mit Verstärkungsring
und Schreibschutzkerbe
inkl. Labelset, unformatiert.



Weitere Angebote auf der Rückseite!

64'er PROGRAMMSERVICE



Sie suchen packende Spiele, hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.
Wenn Sie Fragen zu den Programmen in unserem Angebot haben, rufen Sie uns an:
Telefon (089) 4613-640

Bestellungen bitte *nur gegen Vorauskasse* an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0.
Schweiz:
Markt&Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440550.
Österreich:
Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0; Microcomput-ique, E. Schiller, Göglstraße 17, A-3500 Krems, Telefon (02732) 7493; MES-Versand, Postfach 15, A-3485 Haitzendorf; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196.
Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung der Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die abgedruckte Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

64'er-Sonderheft 44: Top-Anwendungen für alle C128-Besitzer

Gredi: Nutzen Sie mit diesem Zeichenprogramm den 80-Zeichen-Bildschirm Ihres C128 voll aus. Für Textfunktionen ist ein Zeichensatzeditor integriert. Bilder im PrintFox-Format lassen sich problemlos in das Gredi-Format konvertieren. Drucker-Voraussetzung: Epson-Kompatible. **Dispo 128:** Die komfortable Verwaltung einer umfangreichen Diskettensammlung ist für jeden Computerbesitzer unentbehrlich. Der besondere Vorteil von »Dispo 128«: Das Programm erkennt automatisch das verwendete Diskettenformat, und Sie behalten die Übersicht über alle Disketten im 1540/1571/1581-Format. **Flowchart:** Das Werkzeug für strukturiertes Programmieren: Entwerfen Sie am Bildschirm das Flußdiagramm, »Flowchart« generiert daraus automatisch das entsprechende Basic-Programm. **Floppy Support:** Dieses leistungsfähige Disketten-Utility für die Floppy 1570/1571 erlaubt umfangreiche Manipulationen. Direkten Zugriff auf die Disketten erhalten Sie durch den eingebauten Diskettenmonitor, der die Vielzahl der mächtigen Funktionen abrundet. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des 64'er-Sonderhefts 44 mit einem Diskettensymbol gekennzeichnet sind.
Diskette für C128

Bestell-Nr. 15944 **DM 29,90*** sFr 24,90*/öS 299,–*

64'er-Sonderheft 42: Einsteiger-Paket:

Text II: Die besondere Textverarbeitung mit 80 Zeichen pro Zeile. Wordwrapping, Formatieren im Blocksatz, deutsche Tastatur und flexible Druckersteuerung gehören zu diesem sehr schnellen Textprogramm (geeignet nur für Monitor-Besitzer). **SMON:** Dringen Sie mit diesem leistungsfähigen Speichermonitor in die Tiefen Ihres Computers vor. Der integrierte Diskettenmonitor gibt volle Kontrolle über die Floppystation. **Master-Copy Plus:** Bei Unmengen von Software für den C64 tut ein Kopierprogramm not. Genau hier setzt unser Programm Master-Copy Plus an. **Checksummer V3 und MSE:** Wollen Sie Listings abtippen? Zwei Eingabehilfen, die das fehlerfreie Abtippen von Programmen wesentlich erleichtern. Die Anleitungen zu den einzelnen Programmen befinden sich auf der Diskette und können mit dem Programm Text II eingelesen werden.

Bestell-Nr. 15942 **DM 19,90*** (sFr 17,–*/öS 199,–*)

64'er-Sonderheft 41: Floppy optimal genutzt

Disc-Wizard: Dieses außergewöhnliche Disketten-Utility ist für jeden Floppy-Besitzer ein unentbehrliches Hilfsmittel. Es enthält unter anderem einen komfortablen Diskettenmonitor und eine Sortierfunktion für Directory-Einträge. Der Disc-Wizard macht Sie zum Zaubermeister über Ihre Disketten. **Disc-Basic:** Mit 33 neuen Befehlen gehören umständliche Programme zur Bedienung der Floppy der Vergangenheit an. Das Anlegen einer relativen Datei erledigt diese Spracherweiterung zum Beispiel mit einem einzigen Befehl. Die neuen Befehle machen Schluß mit dem bisherigen Floppy-Kauderwelsch. **Diskprint:** Viele pfiffige Details lassen die Arbeit mit dieser Diskettenverwaltung zu einem wahren Vergnügen werden. Bis zu 250 Disketten werden von dem Programm verwaltet. Die integrierten Editierfunktionen sorgen für Übersichtlichkeit am Bildschirm und beim Ausdruck. **64'er-DOS V4 und Autostart:** Auch Bewährtes läßt sich noch verbessern: Der Floppys-Speeder mit integriertem Monitor SMON läßt sich jetzt in jeden C64 und in jede Floppy einbauen. Mit dem Autostart-System werden Programme sofort nach dem Einschalten automatisch geladen und gestartet. (Die Programme müssen auf EPROM gebrannt werden.) **Disk-Tape-Backup:** Aufatmen für jeden Datasetten-Besitzer: Mit diesem Programm wird das Kopieren von Floppy auf Datasette zum Kinderspiel. Ein eingebauter Schnelllader sorgt dabei für Komfort. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des 64'er-Sonderhefts 41 mit einem Diskettensymbol gekennzeichnet sind.
Eine Diskette

Bestell-Nr. 15941 **DM 19,90*** (sFr 17,–*/öS 199,–*)

64'er-Sonderheft 39: Das Komplett-Paket für professionelles Desktop Publishing

Giga-Publish: Ideal für Vereine oder Schülerzeitungen: Giga-Publish ist ein professionelles Desktop-Publishing-Programm der Spitzenklasse. Gestalten Sie Ihre Texte und Grafiken zu einem perfekten Layout. Acht Seiten mit zehn verschiedenen Zeichensätzen können Sie gleichzeitig bearbeiten. **Mastertext:** Dieses Textverarbeitungsprogramm der Spitzenklasse bietet durch seine einfache Menüsteuerung einen hohen Bedienungskomfort. Es ist die ideale Ergänzung für Giga-Publish. **Master-Address:** Adreßverwaltung, Serienbriefe, Adreßaufkleber und Etiketten sind die Domäne von Master-Address. Es ist optimal auf Mastertext abgestimmt und bildet mit diesem ein leistungsfähiges Software-Paket. **Master-Spell:** Nie wieder Tipp- und Flüchtigkeitsfehler: Eine automatische Rechtschreibprüfung ist für alle unentbehrlich, die mit Mastertext arbeiten. **Hi-Eddi:** Dieses komfortable und leistungsfähige Zeichen- und Konstruktionsprogramm macht es einfach, Grafiken mit Joystick oder Maus zu erstellen. Mit Mastertext und Hi-Eddi verfügen Sie über das ideale Gespann, Giga-Publish als DTP-Programm optimal zu nutzen. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des 64'er-Sonderhefts 39 mit einem Diskettensymbol gekennzeichnet sind.
Eine Diskette

Bestell-Nr. 15939 **DM 19,90*** sFr 17,–*/öS 199,–*

64'er-Sonderheft 38: Komplettes Einsteiger-Paket

Diskette 1, Grafik total:

Eine Auswahl faszinierender Bilder, die zeigen, welche grafischen Fähigkeiten im C64 stecken. **Paint Magic:** Ein tolles Grafikprogramm, mit dem Sie in kürzester Zeit wunderschöne farbige Grafiken und Bilder auf den Monitor zaubern. **Alpha Drummer:** Dieses Schlagzeug-Programm liefert 24 perfekte Sounds. Wer möchte, kann beliebige Rhythmen erzeugen oder eigene Sounds digitalisieren. **Sounds zum Genießen:** Entlocken Sie dem Sound-Chip Ihres C64 Musik, die Sie vom Hocker reißt. Von Klassik bis Pop – für jeden ist etwas dabei.

Bestell-Nr. 15938 **DM 19,90*** sFr 17,–*/öS 199,–*

Diskette 2 Leichter lernen mit dem Computer:

Jetzt ist der Frust beim öden Pauken vorbei. Vier Programme helfen dabei: Der **»Vokabeltrainer«** bringt mehr Spaß beim Englischlernen. Ein Übungsprogramm zum **»Bruchrechnen«** erleichtert Schülern, diese gefürchtete Hürde zu überwinden. **»Lateinische Deklinationen«** greift auch bei dieser Fremdsprache unter die Arme. **»CAT«** bringt den »Kleineren« die Grundrechenarten mit grafischer Unterstützung näher. **Springvogel:** Helfen Sie dem Springvogel beim Eiersammeln: ein Spiel für geschickte Hände. Mit dem eingebauten Editor können Sie eigene Spielstufen erstellen und die Schwierigkeit Ihren Bedürfnissen anpassen. **Pro-Disk:** Mit dieser professionellen Diskettenverwaltung behalten Sie stets den Überblick über Ihre Programmsammlung.

Bestell-Nr. 16938 **DM 19,90*** sFr 17,–*/öS 199,–*

Aktionspreis: Alle Programme, die im Inhaltsverzeichnis des 64'er-Sonderheftes 38 mit einem Diskettensymbol gekennzeichnet sind (inkl. der Programme aus Diskette 1 und 2) erhalten Sie auf zwei Disketten
Zwei Disketten

Bestell-Nr. 17938 **DM 29,90*** sFr 24,90*/öS 299,–*

*Unverbindliche Preisempfehlung

Wichtig: Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,– können Sie Software-Disketten Ihrer Wahl aus dem Programmservice Angebot im Wert von DM 180,– bestellen – egal, ob diese DM 19,90, DM 29,90 oder DM 89,– kosten. Sie sparen DM 30,–!

Das Super-Software-Angebot finden Sie in den Zeitschriften

Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST Magazin, PC Magazin, Happy-Computer.

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken!

Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingehefteten Zahlkarte direkt beim Verlag bestellen. Kennwort: »Super-Software-Scheckheft«, Bestell-Nr. W156

```
0129              LDA   OBEN       ; RASTERZEILE AUS OBEN
0130              STA   RASTER     ; VORWAEHLEN
0131              LDA   CONTROL1   ; KEIN UEBERTRAG
0132              AND   #$7F
0133              STA   CONTROL1
0134 ;
0135              LDA   #%10000001 ; RASTER-INTERRUPT
0136              STA   IRQMASK    ; WAEHLEN
0137 ;
0138              LDA   #<IRQPLUS  ; ZEIGER SETZEN
0139              STA   IRQVECTORL
0140              LDA   #>IRQPLUS
0141              STA   IRQVECTORH
0142 ;
0143              JSR   MULTINIT   ;  MULTIPLIKATIONSTABELLE
0144 ;                             ;  BERECHNEN
0145              JSR   COVERSPRITES ; SPRITES ZUM VERDECKEN
0146 ;                             ; ERZEUGEN
0147 ;
0148              LDA   #$00       ; LETZTES BYTE DER
0149              STA   $3FFF      ; VIDEOBANK LOESCHEN
0150 ;
0151 WAIT         LDA   RASTER     ; AUF UNTERE RASTERZEILE
0152              CMP   UNTEN      ; WARTEN
0153              BNE   WAIT
0154 ;
0155              CLI              ; IRQ WIEDER ZULASSEN
0156              RTS              ; ... UND ZURUECK
0157 ;
0158 ;***********************************
0159 ;*     NEUE INTERRUPTROUTINE       *
0160 ;***********************************
0161 ;
0162 IRQPLUS      LDA   IRQFLAG    ; IRQ-REGISTER AUSLESEN
0163              STA   IRQFLAG    ; UND SOMIT LOESCHEN
0164 ;
0165              LDA   RASTER     ; RASTERZEILE BESTIMMEN
0166              CMP   UNTEN      ; UND VERZWEIGEN
0167              BCS   JMPUNTEN
0168              BCC   ISTOBEN
0169 ;
0170 JMPUNTEN     JMP   ISTUNTEN
0171 ;
0172 ;
0173 ;
0174 ;
0175 ;***********************************
0176 ;*     OBERER TEIL DES IRQS        *
0177 ;***********************************
0178 ;
0179 ISTOBEN      JSR   PARAMOBEN  ; VIC-WERTE SETZEN
0180              JSR   VSCROLL    ; VERTIKAL SCROLLEN
0181              JSR   VCONST     ;
0182              JSR   HSCROLL    ; HORIZONTAL SCROLLEN
0183              JSR   HCONST     ;
0184              JSR   JOYSTICK   ; ABFRAGEN
0185              JSR   HARDPREP   ; HARDSCROLL VORBEREITEN
0186 ;
0187              LDA   UNTEN      ; RASTERZEILE SETZEN
0188              STA   RASTER
0189              JMP   IRQNORMAL  ; ... UND ZURUECK
0190 ;
0191 ;***********************************
0192 ;*     PARAMETER SETZEN            *
0193 ;***********************************
0194 ;
0195 PARAMOBEN    LDA   RFARBE1    ; FARBEN FESTLEGEN
0196              STA   RAHMEN
0197              LDA   HFARBE1
0198              STA   HINTERGRUND1
0199 ;
0200              LDA   #$00       ; BILDSCHIRM
0201              STA   CONTROL1   ; ABSCHALTEN
0202 ;
0203              LDA   YSOFT      ; VERTIKALEN SCROLL-WERT
0204              LSR              ; HOLEN
0205              LSR              ; DURCH 32 TEILEN
0206              LSR
0207              LSR
0208              LSR
0209              AND   #$07       ; BITS 3-7 ISOLIEREN
0210              BEQ   NULL       ; AUF NULLGRENZE PRUEFEN
0211              STA   CONTROL1   ; UND INS SCROLLREGISTER
0212 ;                             ; SCHREIBEN
0213 ;
0214 NULL         LDA   XSOFT      ; HORIZONTALER SCROLL-WERT
0215              LSR              ; DURCH 32 TEILEN
0216              LSR
0217              LSR
0218              LSR
0219              LSR
0220              AND   #$07       ; BITS 3-7 ISOLIEREN
0221              ORA   #%11010000 ;
0222              STA   CONTROL2   ; UND SCROLL-WERT SETZEN
0223 ;
0224 ;
0225              LDA   CONTROL1   ; BILDSCHIRM WIEDER
0226              ORA   #$10       ; EINSCHALTEN
0227              STA   CONTROL1
0228 ;
0229              RTS              ; ... UND ZURUECK
0230 ;
0231 ;
0232 ;
0233 ;***********************************
0234 ;*     SOFTSCROLLEN                *
0235 ;***********************************
0236 ;
0237 ; AB HIER WIRD DURCH VERAENDERN
0238 ; DER ENTSPRECHENDEN ZWISCHENSPEICHER
0239 ; DAS SCROLLEN DURCHGEFUEHRT. DER
0240 ; INHALT DER ZWISCHENSPEICHER WIRD
0241 ; DURCH OBIGE ROUTINE IN DIE ZWEI
0242 ; SCROLLREGISTER DES VIC GESCHRIEBEN.
0243 ;
0244 ;***********************************
0245 ;*     VERTIKAL - VARIABEL         *
0246 ;***********************************
0247 ;
0248 VSCROLL      LDA   YSPEED     ; RICHTUNG BESTIMMEN
0249              BMI   OBENSOFT   ; UND VERZWEIGEN
0250              JMP   UNTENSOFT
0251 ;
0252 ;
0253 UNTENSOFT    LDA   YSPEED     ; MOMENTANE GESCHWINDIGKEIT
0254              SEC              ; HOLEN
0255              SBC   YFRICT     ; REIBUNG ABZIEHEN
0256              BCS   BRANCH1    ; BEI STILLSTAND
0257              RTS              ; ZURUECK
0258 BRANCH1      STA   YSPEED     ; GESCHWINDIGKEIT SCHREIBEN
0259              LDA   YSOFT      ; SCROLL-WERT HOLEN
0260              CLC              ; GESCHWINDIGKEIT DAZU
0261              ADC   YSPEED     ; ADDIEREN
0262              STA   YSOFT      ; UND SETZEN
0263              BCS   UNTENHARD  ; BEI UEBERLAUF HARDSCROLL
0264              RTS              ; SONST ZURUECK
0265 ;
0266 OBENSOFT     LDA   YSPEED     ; ANALOG ZU OBIGEM ABLAUF,
0267              CLC              ; DURCH VORZEICHENUMKEHR
0268              ADC   YFRICT     ; RICHTUNGSWECHSEL
0269              BCC   BRANCH2
0270              RTS
0271 BRANCH2      STA   YSPEED
0272              LDA   YSOFT
0273              CLC
0274              ADC   YSPEED
0275              STA   YSOFT
0276              BCC   OBENHARD
0277              RTS
0278 ;
0279 ;***********************************
0280 ;*     VERTIKAL - KONSTANT         *
0281 ;***********************************
0282 ;
0283 VCONST       LDA   YCONST     ; KONSTANTE RICHTUNG
0284              BMI   OBENCONST  ; BESTIMMEN
0285              JMP   UNTENCONST ; UND VERZWEIGEN
0286 ;
0287 ;
0288 UNTENCONST   LDA   YSOFT      ; KONSTANTE BEWEGUNG
0289              CLC              ; ZUM SCROLL-WERT
0290              ADC   YCONST     ; ADDIEREN
0291              STA   YSOFT
0292              BCS   UNTENHARD
0293              RTS
0294 ;
0295 OBENCONST    LDA   YSOFT      ; BZW. ALS NEGATIVE
0296              CLC              ; ZAHL DAVON SUBTRA-
0297              ADC   YCONST     ; HIEREN
0298              STA   YSOFT
0299              BCC   OBENHARD
0300              RTS
0301 ;
0302 ;***********************************
0303 ;*     HARDSCROLL DURCH AENDERN    *
0304 ;*     DES ZEILENZEIGERS           *
0305 ;***********************************
0306 ;
0307 ; HIER KANN EINE ABFRAGE AUF ERREICHTE
0308 ; GRENZEN IM SCROLLSCREEN STEHEN !
0309 ;
0310 ;
0311 OBENHARD     INC   YPOINTER   ; EINE ZEILE HOEHER
0312              RTS
0313 ;
0314 UNTENHARD    DEC   YPOINTER   ; EINE ZEILE TIEFER
0315              RTS
0316 ;
0317 ;
0318 ;
0319 ;***********************************
0320 ;*     HORIZONTAL - VARIABEL       *
0321 ;***********************************
0322 ;
0323 ; DIESER PROGRAMMTEIL FUEHRT ANALOG
0324 ; ZUR OBIGEN ROUTINE DIE SCROLL-
0325 ; BEWEGUNG AUS. DIESE ERFOLGT HIER
0326 ; ALLERDINGS IN HORIZONTALER RICHTUNG.
0327 ;
0328 HSCROLL      LDA   XSPEED     ; RICHTUNG BESTIMMEN
0329              BMI   LINKSSOFT  ; UND VERZWEIGEN
0330              JMP   RECHTSSOFT
0331 ;
0332 RECHTSSOFT   LDA   XSPEED     ; REIBUNG VON
0333              SEC              ; GESCHWINDIGKEIT
0334              SBC   YFRICT     ; ABZIEHEN
0335              BCS   BRANCH3
0336              RTS
0337 BRANCH3      STA   XSPEED     ; GESCHWINDIGKEIT
0338              LDA   XSOFT      ; ZUM SCROLL-WERT
0339              CLC              ; ADDIEREN
0340              ADC   XSPEED
0341              STA   XSOFT
0342              BCS   RECHTSHARD
0343              RTS
0344 ;
0345 LINKSSOFT    LDA   XSPEED     ; WIE OBEN,
0346              CLC              ; ABER IN ANDERE
0347              ADC   XFRICT     ; RICHTUNG DURCH
0348              BCC   BRANCH4    ; VORZEICHENUMKEHR
0349              RTS
0350 BRANCH4      STA   XSPEED
0351              LDA   XSOFT
0352              CLC
0353              ADC   XSPEED
0354              STA   XSOFT
```

Listing 2. (Fortsetzung)

```
0355             BCC   LINKSHARD
0356             RTS
0357 ;
0358 ;***********************************
0359 ;*       HORIZONTAL - KONSTANT     *
0360 ;***********************************
0361 ;
0362 HCONST      LDA   XCONST
0363             BMI   LINKSCONST
0364             JMP   RECHTSCONST
0365 ;
0366 RECHTSCONST LDA   XSOFT      ; KONSTANTE BEWEGUNG
0367             CLC              ; ADDIEREN
0368             ADC   XCONST
0369             STA   XSOFT
0370             BCS   RECHTSHARD
0371             RTS
0372 ;
0373 LINKSCONST  LDA   XSOFT      ; BZW. ALS NEGATIVE
0374             CLC              ; ZAHL SUBTRAHIEREN
0375             ADC   XCONST
0376             STA   XSOFT
0377             BCC   LINKSHARD
0378             RTS
0379 ;
0380 ;***********************************
0381 ;*     HARDSCROLL DURCH AENDERN    *
0382 ;*     DES SPALTENZEIGERS          *
0383 ;***********************************
0384 ;
0385 ; AUCH HIER KANN EINE ABFRAGE AUF
0386 ; GRENZEN IM SCROLLSCREEN STEHEN !
0387 ;
0388 LINKSHARD   INC   XPOINTER   ; EINE SPALTE
0389             RTS              ; NACH LINKS
0390 ;
0391 RECHTSHARD  DEC   XPOINTER   ; BZW. NACH RECHTS
0392             RTS
0393 ;
0394 ;
0395 ;***********************************
0396 ;*     JOYSTICK ABFRAGEN           *
0397 ;***********************************
0398 ;
0399 JOYSTICK    LDA   JOYPORT2   ; BITS DURCH
0400             ROR              ; ROTIEREN
0401             BCC   JOYOBEN    ; UEBERPRUEFEN
0402             ROR              ; UND VERZWEIGEN
0403             BCC   JOYUNTEN
0404 ;
0405 HORIZONTAL  LDA   JOYPORT2
0406             ROR
0407             ROR
0408             ROR
0409             BCC   JOYRECHTS
0410             ROR
0411             BCC   JOYLINKS
0412 ;
0413             RTS
0414 ;
0415 ;
0416 ;***********************************
0417 ;*     DER JOYSTICK-BEWEGUNG ENT-  *
0418 ;*     SPRECHEND GESCHWINDIGKEIT   *
0419 ;*     VERAENDERN                  *
0420 ;***********************************
0421 ;
0422 ;
0423 ;***********************************
0424 ;*     JOYSTICK NACH OBEN          *
0425 ;***********************************
0426 ;
0427 JOYOBEN     LDA   YSPEED     ; GESCHWINDIGKEIT
0428             CLC              ; DURCH ADDIEREN DER
0429             ADC   YACCEL     ; BESCHLEUNIGUNG
0430             BVS   END1       ; ERHOEHEN, BEI UEBER-
0431             STA   YSPEED     ; LAUF STOP
0432 END1        JMP   HORIZONTAL
0433 ;
0434 ;***********************************
0435 ;*     JOYSTICK NACH UNTEN         *
0436 ;***********************************
0437 ;
0438 JOYUNTEN    LDA   YSPEED     ; GESCHWINDIGKEIT
0439             SEC              ; IN ANDERE RICHTUNG
0440             SBC   YACCEL     ; ERHOEHEN BZW.
0441             BVS   END2       ; ERNIEDRIGEN UND
0442             STA   YSPEED     ; UMKEHREN
0443 END2        JMP   HORIZONTAL
0444 ;
0445 ;
0446 ;***********************************
0447 ;*     JOYSTICK NACH RECHTS        *
0448 ;***********************************
0449 ;
0450 JOYRECHTS   LDA   XSPEED     ; ANALOGER VORGANG
0451             CLC              ; ZU OBEN, ABER IN
0452             ADC   XACCEL     ; HORIZONTALER
0453             BVS   END3       ; RICHTUNG
0454             STA   XSPEED
0455 END3        RTS
0456 ;
0457 ;***********************************
0458 ;*     JOYSTICK NACH LINKS         *
0459 ;***********************************
0460 ;
0461 JOYLINKS    LDA   XSPEED
0462             SEC
0463             SBC   XACCEL
0464             BVS   END4
0465             STA   XSPEED
0466 END4        RTS
0467 ;
0468 ;
```

```
0469 ;
0470 ;***********************************
0471 ;*      HARDSCROLLEN VORBEREITEN   *
0472 ;***********************************
0473 ;
0474 ; DIESER PROGRAMMTEIL SETZT MIT HILFE
0475 ; DES ZEILEN- UND SPALTENZEIGERS DAS
0476 ; BILDSCHIRMFENSTER AN DIE RICHTIGE
0477 ; STELLE IM SCROLLSCREEN UND SORGT
0478 ; SO FUER DIE ZEICHENWEISE SCROLL-
0479 ; BEWEGUNG.
0480 ; DIES GESCHIEHT DURCH MODIFIKATION DER
0481 ; ADRESSEN DER HARDSCROLLROUTINE.
0482 ;
0483 HARDPREP    LDY   YPOINTER   ; ZEILENZIEGER HOLEN
0484             LDX   #$00
0485 ;
0486             LDA   LINES      ; SCHLEIFE ENTSPRECHEND
0487             STA   COUNTER    ; DER ANZAHL DER BILD-
0488 ;                             ; SCHIRMZEILEN DURCH
0489 ;
0490 PREPLOOP    CLC
0491             LDA   LOW,Y      ; ZEIGER ZUR ADRESSE AUS
0492             ADC   XPOINTER   ; MULTIPLKATIONSTABELLE
0493             STA   HARDLOOP+1,X ; PLUS DER ANFANGSADRESSE
0494 ;                             ; DES SCROLLSCREENS
0495             LDA   HIGH,Y     ; UND ZUR HARDSCROLLROUTINE
0496             ADC   #>TABELLE  ; SCHREIBEN
0497             STA   HARDLOOP+2,X
0498             INX              ; 6 BYTE ABSTAND
0499             INX              ; VON EINER
0500             INX              ; ADRESSE ZUR NAECHSTEN
0501             INX
0502             INX
0503             INX
0504             INY
0505             DEC   COUNTER    ; SCHON ALLE ZEILEN ?
0506             LDA   COUNTER
0507             BNE   PREPLOOP
0508 ;
0509             RTS              ; ... DANN ZURUECK
0510 ;
0511 ;
0512 ;
0513 ;
0514 ;
0515 ;***********************************
0516 ;*     UNTERER TEIL DES IRQS       *
0517 ;***********************************
0518 ;
0519 ISTUNTEN    JSR   PARAMUNTEN ; VIC-WERTE SETZEN
0520             JSR   HARDSCROLL ; AUSFUEHREN
0521 ;
0522             LDA   OBEN       ; OBERE RASTERZEILE
0523             STA   RASTER
0524             JMP   IRQABSCHLUSS ; ... UND ZURUECK
0525 ;
0526 ;***********************************
0527 ;*     PARAMETER SETZEN            *
0528 ;***********************************
0529 ;
0530 PARAMUNTEN  LDA   RFARBE2    ; FARBEN MIT
0531             NOP              ; VERZOEGERN SETZEN,
0532             NOP              ; UM FLACKERN ZU
0533             NOP              ; VERHINDERN
0534             NOP
0535             NOP
0536             STA   RAHMEN
0537             LDA   HFARBE2
0538             STA   HINTERGRUND1
0539 ;
0540             LDX   #$0D       ; SPRITE-REIHE
0541             LDA   #$D0       ; ZUM VERDECKEN
0542 UKOOR       STA   VIC,X      ; HOLEN
0543             DEX
0544             DEX
0545             BPL   UKOOR
0546 ;
0547             LDA   #%01100000 ; UEBERTRAG SETZEN
0548             STA   SX255
0549 ;
0550             LDA   #$02       ; SONSTIGE
0551             STA   CONTROL1   ; VIC-PARAMETER
0552             LDA   #$D0
0553             STA   CONTROL2
0554 ;
0555             RTS
0556 ;
0557 ;***********************************
0558 ;*     HARDSCROLLEN AUSFUEHREN     *
0559 ;***********************************
0560 ;
0561 HARDSCROLL  LDY   #$27       ; SCHLEIFENDURCHLAUEFE
0562 ;
0563             LDA   #$36       ; SPEICHER UNTER
0564             STA   $01        ; BASIC-ROM EIN
0565 ;
0566 HARDLOOP    LDA   TABELLE,Y ; HARDSCROLL-SCHLEIFE
0567             STA   SCREEN,Y
0568             LDA   TABELLE+100,Y ; HOLT AUS DEM SPEICHER
0569             STA   SCREEN+40,Y ; DEN DURCH DIE ZEIGER
0570             LDA   TABELLE+200,Y ; FESTGELEGTEN BEREICH
0571             STA   SCREEN+80,Y ; UND SCHREIBT IHN AUF
0572             LDA   TABELLE+300,Y ; DEN BILDSCHIRM
0573             STA   SCREEN+120,Y
0574             LDA   TABELLE+400,Y
0575             STA   SCREEN+160,Y ; DIE QUELLADRESSEN WERDEN
0576             LDA   TABELLE+500,Y ; IM OBEREN TEIL DES IRQS
0577             STA   SCREEN+200,Y ; ANHAND DER ZEIGER
0578             LDA   TABELLE+600,Y ; MODIFIZIERT
0579             STA   SCREEN+240,Y
```

```
0580                LDA   TABELLE+700,Y
0581                STA   SCREEN+280,Y
0582                LDA   TABELLE+800,Y
0583                STA   SCREEN+320,Y
0584                LDA   TABELLE+900,Y
0585                STA   SCREEN+360,Y
0586                LDA   TABELLE+1000,Y
0587                STA   SCREEN+400,Y
0588                LDA   TABELLE+1100,Y
0589                STA   SCREEN+440,Y
0590                LDA   TABELLE+1200,Y
0591                STA   SCREEN+480,Y
0592                LDA   TABELLE+1300,Y
0593                STA   SCREEN+520,Y
0594                LDA   TABELLE+1400,Y
0595                STA   SCREEN+560,Y
0596                LDA   TABELLE+1500,Y
0597                STA   SCREEN+600,Y
0598                LDA   TABELLE+1600,Y
0599                STA   SCREEN+640,Y
0600                LDA   TABELLE+1700,Y
0601                STA   SCREEN+680,Y
0602                LDA   TABELLE+1800,Y
0603                STA   SCREEN+720,Y
0604                LDA   TABELLE+1900,Y
0605                STA   SCREEN+760,Y
0606                LDA   TABELLE+2000,Y
0607                STA   SCREEN+800,Y
0608                DEY
0609                BMI   LOOPEXIT
0610                JMP   HARDLOOP
0611 ;
0612  LOOPEXIT      LDA   #$37       ; SPEICHER UNTER
0613                STA   $01        ; BASIC-ROM AUS
0614 ;
0615 ;
0616                RTS
0617 ;
0618 ;
0619 ;
0620 ;***********************************
0621 ;*     EINMALIG AUFGERUFENE        *
0622 ;*     INIT-ROUTINEN               *
0623 ;***********************************
0624 ;
0625 ;***********************************
0626 ;*     MULTIPLIKATIONSTABELLE      *
0627 ;*     BERECHNEN
0628 ;***********************************
0629 ;
0630 ;
0631  MULTINIT      LDA   BREITE     ; AUSDDEHUNG DES
0632                STA   XPOINTER   ; SCROLLSCREENS ALS
0633                LDA   HOEHE      ; FAKTOREN
0634                STA   YPOINTER
0635 ;
0636                LDY   #$00       ; 1.FAKTOR FUER
0637                STY   $28        ; MULTIPLIKATION
0638                LDY   #$00
0639                STY   $29
0640 ;
0641  MULTLOOP      LDX   XPOINTER   ; 2.FAKTOR FUER
0642                STX   $71        ; MULTIPLIKATION
0643                LDX   #$00
0644                STX   $72
0645 ;
0646                JSR   UMULT      ; MULTIPLIZIEREN
0647                TXA              ; ERGEBNIS
0648                PHA              ; RETTEN
0649                TYA
0650                PHA
0651 ;
0652                LDX   $28        ; ERGEBNIS IN TABELLE
0653                PLA              ; SCHREIBEN
0654                STA   HIGH,X
0655                PLA
0656                STA   LOW,X
0657 ;
0658                INC   $28        ; ZAEHLER ERHOEHEN
0659                LDX   $28
0660                CPX   XPOINTER
0661                BNE   MULTLOOP   ; NEIN, DANN NOCHMAL
0662 ;
0663                LDA   #$15
0664                STA   XPOINTER
0665                LDA   #$05
0666                STA   YPOINTER
0667                RTS              ; ... UND ZURUECK
0668 ;
0669 ;
0670 ;***********************************
0671 ;*     SPRITEREIHE ZUM VERDECKEN   *
0672 ;*     UND UFO ERZEUGEN            *
0673 ;***********************************
0674 ;
0675  COVERSPRITES  LDX   #$40       ; BLOCKMUSTER
0676                LDA   #$FF       ; ERZEUGEN
0677  FILL          STA   BLOCK,X
0678                DEX
0679                BPL   FILL
0680 ;
0681                LDX   #$40       ; UFO-MUSTER
0682  PATTERN       LDA   UFO,X      ; HOLEN
0683                STA   BLOCK+$40,X
0684                DEX
0685                BPL   PATTERN
0686 ;
0687                LDX   #$0F       ; KOORDINATEN
0688  KOOR          LDA   SKOOR,X    ; HOLEN UND
0689                STA   VIC,X      ; SETZEN
0690                DEX
0691                BPL   KOOR
0692 ;
0693                LDX   #$06       ; FARBEN DER SPRITES
0694                LDA   HFARBE2    ; SCHREIBEN
0695  COLOR         STA   SPRITEOCOL,X
0696                DEX
0697                BPL   COLOR
0698 ;
0699                LDA   #$01
0700                STA   SPRITEOCOL+7
0701 ;
0702                LDX   #$06       ; BLOCKZEIGER
0703                LDA   #$0D       ; SETZEN
0704  MUSTER        STA   SPOINTER,X
0705                DEX
0706                BPL   MUSTER
0707 ;
0708                LDA   #$0E
0709                STA   SPOINTER+7
0710 ;
0711                LDA   #$FF       ; RESTLICHE
0712                STA   SPRITESAN  ; PARAMERTER
0713                STA   SXEXPAND
0714                LDA   #$80
0715                STA   SYEXPAND
0716 ;
0717                RTS
0718 ;                                           ; ... UND ZURUECK
0719 ;
0720 ;***********************************
0721 ;*     SPRITE-KOORDINATEN          *
0722 ;***********************************
0723 ;
0724  SKOOR         .BYTE $18,$91,$48,$91
0725                .BYTE $78,$91,$A8,$91
0726                .BYTE $D8,$91,$08,$91
0727                .BYTE $38,$91,$A0,$6B
0728                .BYTE $E0
0729 ;
0730 ;
0731 ;***********************************
0732 ;*     SPRITE-MUSTER (UFO)         *
0733 ;***********************************
0734 ;
0735  UFO           .BYTE $00,$00,$00,$00,$00,$00,$00,$00
0736                .BYTE $00,$00,$00,$00,$00,$00,$00,$01
0737                .BYTE $FF,$80,$02,$00,$40,$04,$00,$20
0738                .BYTE $04,$00,$20,$04,$00,$20,$06,$00
0739                .BYTE $60,$3F,$81,$FC,$FF,$FF,$FF,$FD
0740                .BYTE $FF,$BF,$3F,$BD,$FC,$0F,$FF,$F0
0741                .BYTE $00,$FF,$00,$00,$00,$00,$00,$00
0742                .BYTE $00,$00,$00,$00,$00,$00,$00,$00
0743 ;
0744 ;***********************************
0745 ;*     TABELLE MIT PARAMETERN      *
0746 ;***********************************
0747 ;
0748  RFARBE1       .BYTE $01        ; RAHMENFARBE OBEN
0749  HFARBE1       .BYTE $00        ; HINTERGRUNDFARBE OBEN
0750  RFARBE2       .BYTE $06        ; RAHMENFRABE UNTEN
0751  HFARBE2       .BYTE $06        ; HINTERGRUNDFARBE UNTEN
0752  OBEN          .BYTE $2D        ; GRENZEN FUER RASTER-
0753  UNTEN         .BYTE $D0        ; INTERRUPT
0754  BREITE        .BYTE $80        ; AUSDEHNUNG DES
0755  HOEHE         .BYTE $FF        ; SCROLLSCREENS
0756  LINES         .BYTE $15        ; ZEILEN IM BILD
0757  XACCEL        .BYTE $04        ; BESCHLEUNIGUNG
0758  YACCEL        .BYTE $04        ;
0759  XFRICT        .BYTE $01        ; REIBUNG
0760  YFRICT        .BYTE $01        ;
0761  XCONST        .BYTE $00        ; KONSTANTE BEWEGUNG
0762  YCONST        .BYTE $00
0763 ;
0764 ;***********************************
0765 ;*     TABELLE MIT VARIABLEN       *
0766 ;***********************************
0767 ;
0768  XPOINTER      .BYTE $40        ; POSITION IM
0769  YPOINTER      .BYTE $10        ; SCROLLSCREEN
0770  COUNTER       .BYTE $00        ; ZAEHLER
0771  XSPEED        .BYTE $00        ; MOMENTANE
0772  YSPEED        .BYTE $00        ; GESCHWINDIGKEIT
0773  XSOFT         .BYTE $00        ; SCROLL-WERT
0774  YSOFT         .BYTE $00
0775 ;
0776 ;***********************************
0777 ;*     MULTIPLIKATIONSTABELLE      *
0778 ;***********************************
0779 ;
0780                *=    $CE00
0781  LOW           *=    *+256
0782  HIGH          *=    *+256
0783 ;
```

Listing 2. (Schluß)

GRAFIK IN DER

Das gab es noch nie auf dem C 64: Eine Befehlserweiterung, mit der Sie auf dem Bildschirm in drei Dimensionen zeichnen können. Lassen Sie sich verzaubern von den vielen Möglichkeiten, die dieses Programm Ihnen bietet!

Das Programm »Prograf« ist eine Basic-Erweiterung, die neue Befehle zum Erzeugen von Hires-Grafiken zur Verfügung stellt. Der Schwerpunkt liegt dabei auf mathematischen Funktionen, aber auch der Nicht-Mathematiker wird sich mit dem Programm schnell anfreunden. Das Programm übernimmt automatisch die Umrechnung in die Bildschirmkoordinaten, umständliche Operationen mit Funktionen wie Sinus oder Cosinus entfallen.

Neben den üblichen Standardroutinen wie Punkt setzen, Punkt löschen, Linie zeichnen usw. weist das Programm einige ganz herausragende Fähigkeiten auf. Zum Beispiel dreidimensionale Grafik in Zentralperspektive: Die Funktionen zum Zeichnen von Punkten und Linien verarbeiten drei Koordinaten ebenso korrekt wie zwei, genauso die Routinen für Translation und Rotation.

Die Grafik umfaßt 640 x 400 Punkte. Dies ist viermal soviel wie gewöhnlich und bringt auch auf dem Drucker eine wesentliche Qualitätssteigerung. Auf dem Bildschirm ist die gesamte Grafik als Übersicht in einer Auflösung von 320 x 200 Punkten zu sehen.

Die Bearbeitung der Grafik mit dem Joystick oder über die Tastatur ist kein Problem: Mit dem Joystick kann gezeichnet, schattiert, gelöscht werden, mit der Tastatur läßt sich jedes Zeichen beider Systemzeichensätze direkt in die Grafik einbauen. Die Druckroutinen wurden durch Interpolation optimiert: Die hohe Auflösung der Grafik wird nochmals gesteigert, so daß der Matrixdrucker mit maximaler Dichte zeichnet. Achtung: Das Programm arbeitet jedoch nur mit Epson-kompatiblen Druckern zusammen, mit 7-Nadel-Druckern wie dem MPS 801/803 ist ein Ausdruck nicht möglich, da diese nur 480 Punkte in eine Zeile packen können (hier wird jedoch mit bis zu 640 Punkten pro Zeile gearbeitet).

Trotz des sehr großen Speicherbedarfs für Grafik und Maschinenprogramm bleiben noch etwa 15 KByte für Basic-Programme frei. Das sollte in aller Regel für eigene Programme ausreichen. Zusammen mit dem weiter unten beschriebenen Basic-Programm »Funprint« ergibt sich ein sehr leistungsfähiges System zum Konstruieren und zum Zeichnen beliebiger Funktionen.

Code	Farbe
0	Schwarz
1	Weiß
2	Dunkelrot
3	Türkis
4	Violett
5	Grün
6	Dunkelblau
7	Gelb
8	Orange
9	Braun
10	Hellrot
11	Dunkelgrau
12	Mittelgrau
13	Hellgrün
14	Hellbau
15	Hellgrau

Tabelle 1. Die gültigen Farbcodes

Vor das Vergnügen haben die Götter nun mal leider den Schweiß des Abtippens gesetzt. Geben Sie bitte das Programm »Prograf« (Listing 1) mit dem MSE ein. Hinweise dazu finden Sie auf Seite 159. Scheuen Sie sich nicht, dieses etwas längere Listing abzutippen, Sie werden dafür mit einem Grafikprogramm der Oberklasse belohnt. Nach dem Laden mit dem Befehl

```
LOAD"PROGRAF",8,8
```

belegt das Programm den Bereich von \$4360 bis \$5C00 (dezimal 17248 bis 23552), die sichtbare Grafik mit Farbspeicher geht von \$5C00 bis \$7FFF, die eigentliche hochauflösende Grafik findet sich teilweise unter den ROMs im Speicherbereich von \$8000 bis \$FFFF. Sie sollten nach dem Laden im Direktmodus sicherheitshalber den Befehl

```
NEW
```

eingeben, damit alle Zeiger wieder richtig gestellt werden. Der gesamte Speicherbereich ab \$4360 muß außerdem vor dem Überschreiben durch Basic-Variablen und Programme geschützt werden. Dies geschieht in der ersten Zeile eines Basic-Programms durch

```
2 POKE 56,67:CLR
```

Die Anzahl der freien Bytes beträgt jetzt noch 15103 (14,7 KByte). Die Erweiterung kann übrigens auch von einem

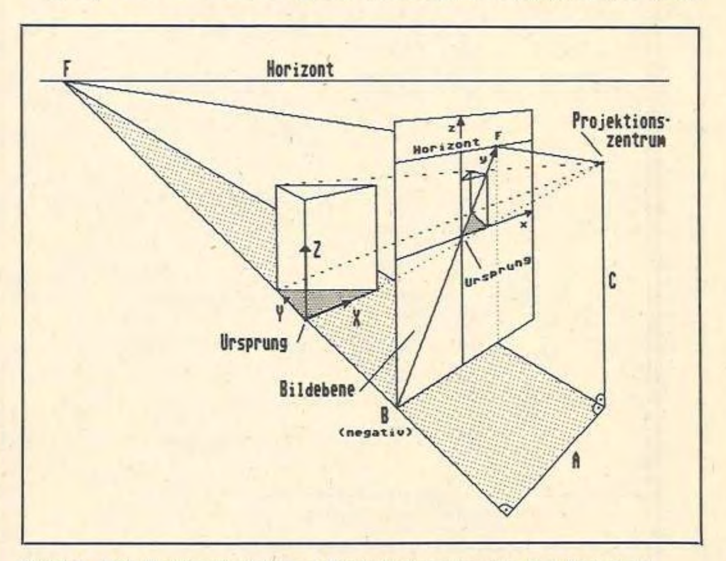


Bild 1. Die Bedeutung des Projektionszentrums bei der Zentralprojektion wird durch die Skizze verdeutlicht

Basic-Programm aus nachgeladen werden, der entsprechende Befehl sieht so aus:

```
1 A=A+1:IF A=1 THEN LOAD "PROGRAF",8,8
```

Die Variable A wird zum Schutz eingesetzt, damit das File nicht mehrmals nachgeladen wird. Nach jedem LOAD startet nämlich das Basic-Programm wieder. Eine andere Möglichkeit finden Sie in den Listings 2 und 3. Nun können die neuen Befehle eingeschaltet werden. Dazu dient der Befehl

```
POKE 777,67
```

3. DIMENSION

Sollten Probleme auftreten (sehr unwahrscheinlich), können Sie die Erweiterung natürlich ohne weiteres auch wieder abschalten. Ein einfacher

```
POKE 777,167
```

genügt da schon.

»Prograf« ist in der vorliegenden Version für Epson-kompatible Drucker geschrieben. Die Anpassung an andere Druckertypen kann mit dem Programm »PG.PRINTER« (Listing 2, bitte mit dem Checksummer eingeben, siehe Seite 159) erfolgen. Dieses Programm wird wie folgt angewendet: Nach dem Abtippen speichern Sie es sicherheitshalber auf einer Leerdiskette. Jetzt geben Sie

```
LIST
```

ein. Das Programm enthält einige DATA-Zeilen, in denen die benötigten Druckercodes zu finden sind. Fügen Sie nun, gegebenenfalls unter Zuhilfenahme des Druckerhandbuchs, die Codes Ihres Druckers ein. In der letzten DATA-Zeile wird der neue Filename eingetragen, den das geänderte Druckprogramm erhalten soll. Jetzt wird Listing 2 mit

```
RUN
```

gestartet. Nach Beantworten der Sicherheitsabfrage mit <J> und <RETURN> wird Listing 1 automatisch nachgeladen, modifiziert und kann gespeichert werden. Dazu legen Sie eine Diskette ein und bestätigen die Abfrage. Jetzt wird das Druckprogramm unter dem neuen Namen gespeichert und kann normal geladen und benutzt werden. Da die Codes leider auf 7-Nadel-Druckern wie dem MPS 801 nicht entsprechend verfügbar sind, ist ein Ausdruck auf diesen Druckern nicht möglich.

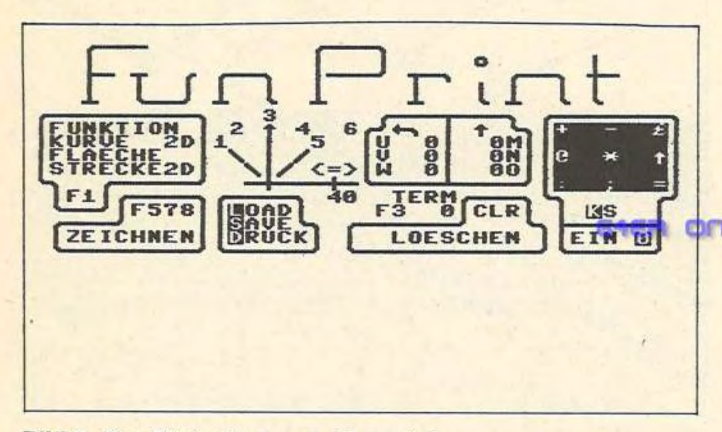


Bild 2. Der Bildschirm von »Funprint«

Dazu gehört der MPS 803 ebenso wie die alten Commodore-Drucker des Typs VC 1515 und VC 1525 sowie der Seikosha GP-100 VC und natürlich alle Drucker, die mit einem speziellen Interface einen MPS 801 simulieren. Allerdings können Sie speziell bei diesen Druckern wenigstens die sichtbare 320 x 200-Punkte-Grafik mit einem Hardcopyprogramm wie »Uniprint« (in 64'er, 4/88 und Sonderheft 32) oder dem »Hardmaker« zu Papier bringen.

Als möglicher Programmkopf für eigene Basic-Programme, die »Prograf« verwenden, kann der »PG.LOADER« (Listing 3, bitte mit dem Checksummer eingeben) dienen. Hier sehen Sie eine Möglichkeit, »Prograf« nachladen zu lassen, falls es sich noch nicht im Speicher befindet.

Jeder Befehl beginnt mit einem Ausrufezeichen und wird wie ein normaler Basic-Befehl eingesetzt. Die Parameter werden grundsätzlich durch ein Komma vom Befehlswort und untereinander getrennt. Parameter, die in der Beschreibung in Klammern stehen, können weggelassen werden (sind optional). Eine Übersicht finden Sie in Tabelle 2.

Wenn Sie übrigens einige der mathematischen Fachausdrücke nicht auf Anhieb verstehen, ist das in keinster Weise ehrenrührig. Wir haben für Sie ein kleines Mathelexikon (Textkasten) vorgesehen, in dem alle Begriffe kurz erklärt sind.

Noch etwas sollten Sie beachten, wenn Sie die Befehle in eigenen Programmen einsetzen wollen. Da die neuen Kommandos nicht in Tokens gewandelt werden (also nicht verkürzt in Basic-Programmen gespeichert werden, sondern im vollen Wortlaut), müssen Sie hinter dem Befehl THEN einen Doppelpunkt setzen, wenn danach ein neuer Befehl folgt. Also nicht

```
IF A = 1 THEN !CLR
```

sondern

```
IF A = 1 THEN : !CLR
```

aber nach wie vor

```
IF A = 1 THEN GOTO 25
```

Befehl, Parameter	Funktion
!CLR	Grafik löschen
!SCN, X	Grafik ausschalten (X=0) oder einschalten (X=1)
!LOC, X, Y (,XU,XO,YU,YO)	Koordinatensystem festlegen
!PLOT X,Y (,Typ)	Punkt in Grafik zeichnen
!LINE X1,Y1,X2,Y2 (,Typ)	Linie zeichnen
!PROZ,A,B,C	Projektionszentrum
!ROT,W1,W2,W3	Rotation festlegen
!TRANS,X0,Y0,Z0	Translation
!ZPLOT,X,Y,Z (,Typ)	3D-Punkt zeichnen
!ZLINE,X1,Y1,Z1,X2,Y2,Z2 (,Typ)	3D-Linie zeichnen
!DEFFN'Name'	Funktion definieren
!DRAW (,Zoomfarbe,Pfeilfarbe)	Grafik mit Joystick und Tastatur verändern
!SAVE"NAME"	Grafik speichern
!LOAD"NAME"	Grafik laden
!COPY,Typ (,Sekundäradresse)	Hardcopy

Tabelle 2. Kurze Übersicht über alle neuen Befehle der Basic-Erweiterung »Prograf«

Kurzinfo: Prograf

Programmart: Basic-Erweiterung für dreidimensionale Zeichnungen
Laden: LOAD "PROGRAF",8,8
Start: Nach dem Laden NEW eingeben. Um das Programm vor Überschreiben zu schützen, geben Sie im Direktmodus »POKE 56,67:CLR« ein. Aktiviert wird es mit POKE 777,67. Ausschalten können Sie es mit POKE 777,167.
Besonderheiten: Das Programm »PG.PRINTER« dient zur Druckeranpassung. Unterstützt werden nur Epson-kompatible Drucker. »PG.LOADER« ist ein Beispiel, wie »Prograf« in einem Basic-Programm nachgeladen wird.
»FUNPRINT« dient zum Experimentieren mit den neuen Befehlen der Erweiterung. Mit dem »DEMOPROGRAMM« lassen sich verschiedene Bilder, die mit »Prograf« gezeichnet werden, auf dem Bildschirm darstellen.
Programmautor: Bertram Hafner

(denn GOTO ist kein neuer Befehl). Nun zur Beschreibung der Prograf-Befehle:

Grafik löschen: !CLR (,Hintergrundfarbe,Zeichenfarbe)

Mit dieser Anweisung wird die Grafik gelöscht. Auf Wunsch können die Farben für den Hintergrund und die Zeichenfarbe angegeben werden. Der Farbspeicher wird dann entsprechend beschrieben. Tabelle 1 enthält die gültigen Farbcodes.

Grafik ein/aus: !SCN,1 oder !SCN,0

Mit dem Wert 1 wird die Grafik eingeschaltet, mit 0 wieder abgeschaltet. Tritt während der Hires-Darstellung ein Fehler im Basic-Programm auf, oder ist das Program beendet,

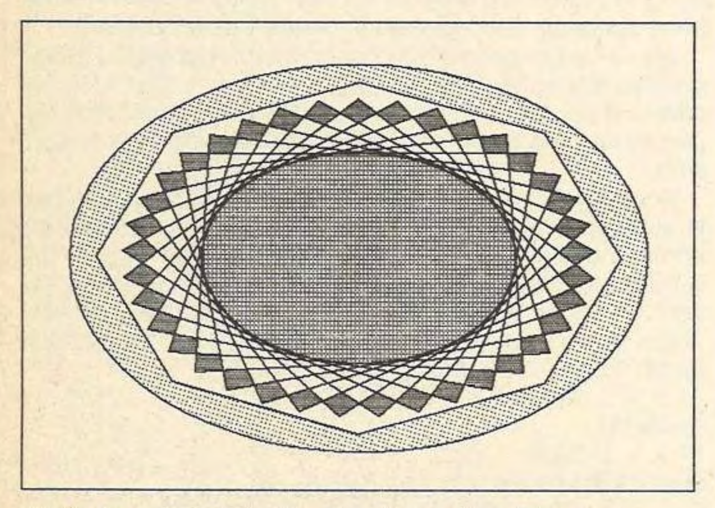


Bild 3. So entwickelt sich schnell aus einer Ellipse ein Vieleck oder eine Rosette

wird automatisch auf die Textdarstellung zurückgeschaltet. Dieser Befehl ist daher im Direktmodus nicht sinnvoll.

Ursprung: !LOC,X,Y (,XU,XO,YU,YO)

Die Locate-Anweisung verschiebt den Koordinatenursprung der Grafik von der linken oberen Ecke des Bildschirms an eine beliebige Stelle (X,Y), die auch außerhalb des Bildschirms liegen darf. Die nächsten vier Werte schränken, wenn sie angegeben sind, den Zeichenbereich auf ein Rechteck innerhalb des Bildschirms ein.

Punkt setzen: !Plot, X,Y (,Typ)

Diese Anweisung setzt einen Punkt (X,Y) in dem durch !LOC definierten Koordinatensystem, sofern der Punkt im festgelegten Zeichenbereich liegt.

```
TYP=0:   Punkt löschen
TYP=1:   Punkt invertieren
TYP=255: Punkt setzen
```

Der Defaultwert ist 255, d. h., wenn Sie keinen Typ angeben, wird der Punkt automatisch gesetzt.

Punkte verbinden: !LINE,X1,Y1,X2,Y2 (,Typ)

Die LINE-Anweisung verbindet die Punkte mit den Koordinaten (X1,Y1) und (X2,Y2) durch eine gerade Linie. Gezeichnet wird nur der Teil, der im definierten Zeichenbereich liegt. Für den Zeichentyp gelten dieselben Vereinbarungen wie bei !PLOT.

Taste	Funktion
<–>	Auswahl des Zeichentyps
<RETURN>	Beenden
<CTRL 1>	Zeichensatz 1
<CTRL 2>	Zeichensatz 2
<CTRL 3>	doppelte Zeichenhöhe
<CTRL 4>	einfache Zeichenhöhe
<CTRL RVSON>	Reverszeichen
<CTRL RVSOFF>	Normalzeichen

Tabelle 3. Die Steuertasten des DRAW-Modus

Projektionszentrum: !PROZ,A,B,C

Diese Anweisung legt das Projektionszentrum (= Betrachterstandort) für die 3D-Darstellung in Zentralperspektive fest (siehe Bild 1). A,B,C sind die drei Koordinaten des Projektionszentrums im Koordinatensystem, das durch !LOC definiert wurde. Die drei Koordinaten liegen immer wie folgt:

```
X-Richtung nach rechts
Y-Richtung in die Tiefe
Z-Richtung nach oben
```

Der Befehl !PROZ wirkt auf die weiter unten beschriebenen Anweisungen !ZPLOT und !ZLINE.

Rotation: !ROT,W1,W2,W3

Die Anweisung !ROT bewirkt beim dreidimensionalen Zeichnen mit !ZPLOT oder !ZLINE zusätzlich eine Drehung um die drei Koordinatenachsen. W1, W2 und W3 sind die drei Eulerschen Winkel im Bogenmaß (siehe Mathelexikon). Der Wert Pi entspricht also dem Winkel 180 Grad,

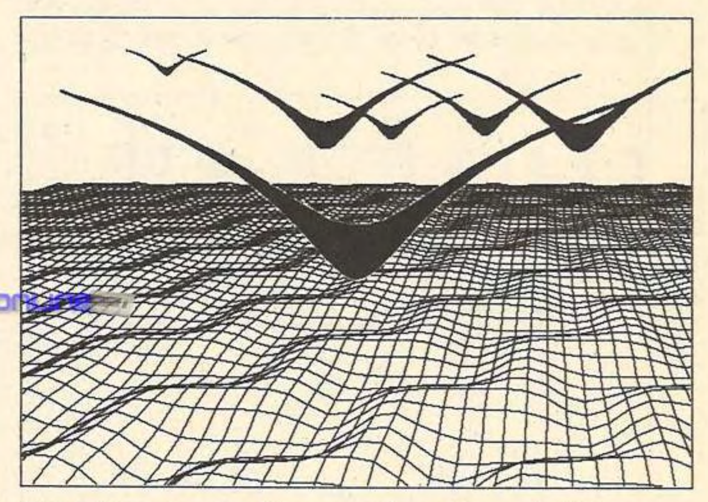


Bild 4. Varianten der Sinus- und Logarithmusfunktion. In Tabelle 4 finden Sie die Koordinaten.

1 Grad = Pi/180. W1 gibt den Winkel der Drehung von der X- in die Y-Richtung (also um die Z-Achse) an. W2 bestimmt die Drehung um die X-Achse, W3 dreht um die Y-Achse. Soll um eine Achse keine Drehung erfolgen, setzen Sie einfach Null für den entsprechenden Winkel ein. Beispiel:

```
!ROT 1.48,0,2.32
```

dreht nur um die X- und Z-Achsen.

Translation: !TRANS,X0,Y0,Z0

Dieser Befehl verschiebt einen Punkt (X,Y,Z) beim Zeichnen mit !ZPLOT oder !ZLINE um die Werte X0,Y0,Z0 im Koordinatensystem, und zwar erst nach der Drehung durch !ROT.

3D-Plot: !ZPLOT,X,Y,Z (,Typ)

Diese Anweisung setzt einen Punkt in Zentralperspektive unter Berücksichtigung von !ROT und !TRANS. Sonst arbeitet der Befehl wie !PLOT. Der Computer leistet hier Schwerstarbeit: Zunächst wird, falls mit !ROT eine Drehung erwünscht ist, diese ausgeführt. Dabei müssen unter Umständen alle drei Koordinaten verändert werden. Jetzt addiert der Computer zu den drei Koordinaten die bei !TRANS definierten Verschiebungen dazu. Die noch dreidimensionalen Koordinaten rechnet er unter Berücksichtigung der bei !PROZ gemachten Angaben in zweidimensionale Bildschirmkoordinaten um (mit Hilfe von trigonometrischen Funktionen) und zeichnet den Punkt in die Grafik ein.

3D-Line: !ZLINE,X1,Y1,Z1,X2,Y2,Z2 (,Typ)
Mit dieser Anweisung werden die beiden Punkte (X1,Y1,Z1) und (X2,Y2,Z2) perspektivisch verbunden. !ROT und !TRANS werden berücksichtigt. Sonst wie !LINE.

Funktionsdefinition: !DEFFN'Name'
Der alte Basic-Befehl DEF FN... ermöglicht bekanntlich eine Funktionsdefinition nur dann, wenn der Funktionsterm bereits im Listing direkt hinter dem Befehl steht. Mit der Routine !DEFFN kann man eine Funktion während des Programmlaufs z. B. mit INPUT definieren. Der neue Befehl

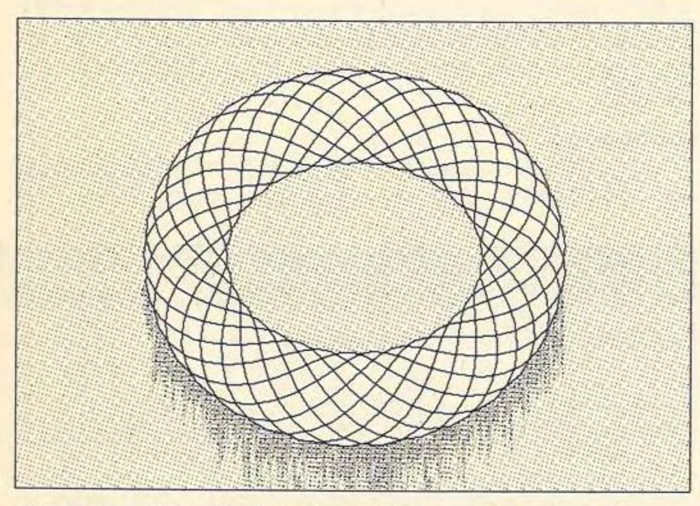


Bild 5. Beachten Sie bei dieser Hardcopy vor allem den Schatten am Rand der Figur

!DEFFN codiert den per INPUT eingegebenen Funktionsterm und baut ihn direkt in das Basic-Listing ein. Gehen Sie dazu wie folgt vor:
1. Im Basic-Programm muß eine Funktion vordefiniert sein.
2. Lesen Sie, z. B. mit INPUT, den neuen Funktionsterm ein. Er steht dann im Basic-Eingabepuffer ab Adresse 512.
3. Aufruf der Routine !DEFFN
   Dazu ein Beispiel:

```
10 DEFFNA(X)=::::::::::::::::::::REM so viele
Doppelpunkte wie möglich als Platzhalter
20 INPUT B$
30 !DEFFNA
```

Nach RUN und LIST sehen Sie die Wirkung: Das Programm hat sich selbst verändert. Anstelle der Doppelpunkte steht jetzt die eingegebene Funktion im Programm.

Joystick/Tastatur: !DRAW (,Zoomfarbe,Pfeilfarbe)
Die !DRAW-Anweisung aktiviert einen Spezialmodus zur Nachbearbeitung der Grafik mittels Joystick (Port 2) und Tastatur. Bei Aufruf wird die Hires-Grafik eingeschaltet. Auf dem Bildschirm erscheint ein beweglicher Pfeil (in der frei wählbaren Pfeilfarbe), der die Funktion des Pinsels bzw. der Schreibmarke übernimmt. Sobald man den Joystick betätigt, wird die Zoom-Funktion eingeschaltet: Der 32 x 32-Punktebereich um den Pfeil wird stark vergrößert dargestellt (in der ebenfalls variablen Zoomfarbe).

Die Bewegung des Pfeils erfolgt mit dem Joystick, den Cursortasten (punktweise) und den Funktionstasten <F5> bis <F8> (in Achterschritten). Mit der Tastatur kann Text direkt in die Grafik getippt werden. Dabei ist der gesamte Zeichensatz des C 64 aktiv. Dies ergibt umfassende Gestaltungsmöglichkeiten, denn auch die Grafiksymbole sind verfügbar. Wird ein Buchstabe vor nicht gelöschten Hintergrund gesetzt, bleibt dieser hinter dem Buchstaben (etwa im Kreis beim großen »O«) erhalten. Nur die Tasten <SPACE> und <DEL> löschen den Hintergrund. Folgende Tastenkombinationen haben eine besondere

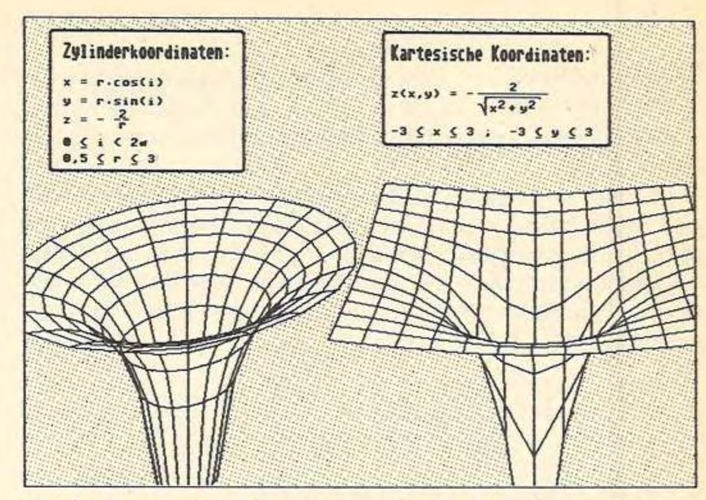
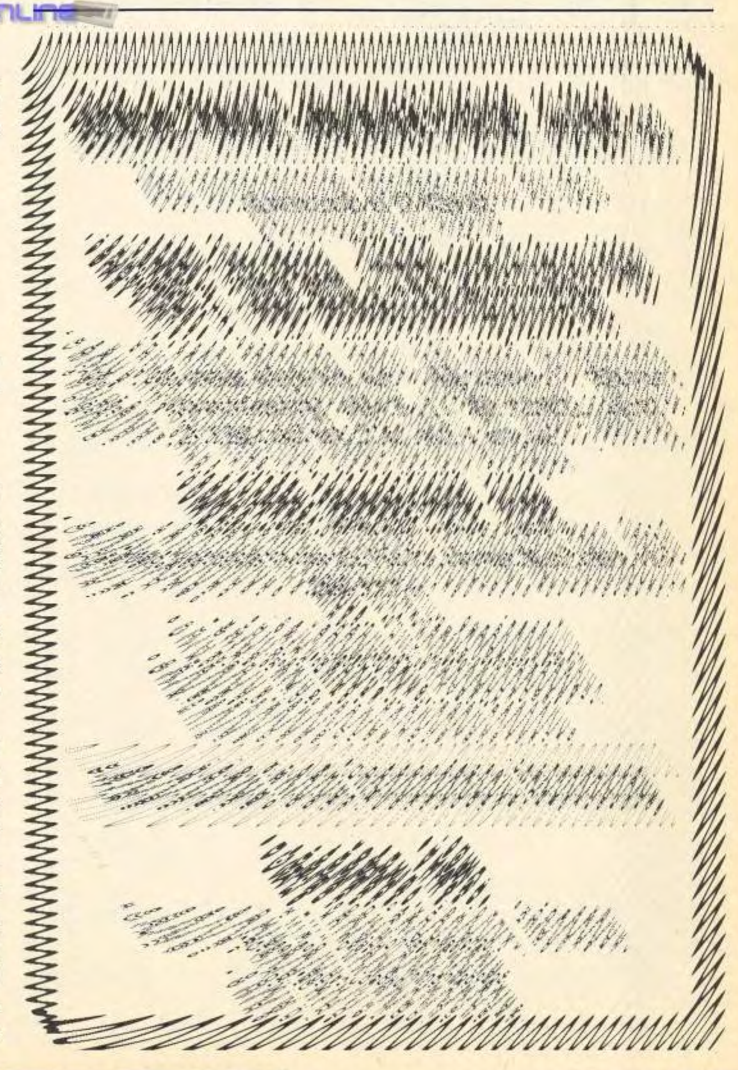


Bild 6. Zweimal der Graph der gleichen Funktion (links: Zylinderkoordinaten, rechts: kartesische Koordinaten)

Schaltfunktion:

<←>:	Auswahl des Zeichentyps (siehe unten).
<RETURN>:	Beenden der DRAW-Anweisung. Der Textmodus wird wieder eingeschaltet, das Programm fährt im Basic-Programm oder im Direktmodus fort.
<CTRL 1>:	Umschalten auf Zeichensatz 1
<CTRL 2>:	Umschalten auf Zeichensatz 2
<CTRL 3>:	Umschalten auf doppelte Zeichenhöhe



<CTRL 4>: Umschalten auf einfache Zeichenhöhe
<CTRL RVSON>: Umschalten auf Reverszeichen
<CTRL RVSOFF>: Umschalten auf Normalzeichen

Diese Tasten finden Sie in Tabelle 3 noch einmal zusammengefaßt.

Mit dem Joystick können Sie den Cursor (Pfeil) innerhalb der Grafik frei bewegen. Drücken Sie den Feuerknopf, wird je nach Zeichentyp ein Punkt invertiert oder eine Sonderaktion ausgeführt: Mit < ← > wählen Sie zwischen verschiedenen Zeichentypen, die unterhalb des Pfeils angezeigt werden. Diese sind:

1. Punkt invertieren
2. eine umschlossene Fläche schraffieren
3. eine umschlossene Fläche schattieren (drei Dichten)
4. eine umschlossene Fläche füllen
5. eine zusammenhängende Fläche löschen
6. eine Linie ziehen
7. eine Linie löschen (durch Markieren von Anfangs- und Endpunkt; ist der Anfangspunkt gelöscht, so wird die ganze Linie gelöscht)
8. eine Linie löschen durch Antippen eines Punktes der Linie

Die Peripherie

Grafik speichern: !SAVE "Name"

Diese Anweisung speichert die 640 x 400-Punktegrafik auf Diskette. Der 32K-Speicherbereich $8000—$FFFF wird vorher auf einfache Weise gepackt: Je mehr Nullbytes die Grafik enthält, um so geringer ist der Platzbedarf auf Diskette.

Grafik laden: !LOAD "Name"

Mit !LOAD wird eine von Prograf gespeicherte Grafik wieder von Diskette geladen und dabei entpackt. Die Grafik

Kleines Mathe-

Achse: Gerade mit besonderer Aufgabe
Abszisse: erste Koordinate eines Punktes, gewöhnlich die (waagerechte) X-Achse. Siehe auch Ordinate
Algorithmus: Rechenregel, Vorschrift zur Berechnung
Argument: Zahl, Wert, Parameter, der in eine Funktion eingesetzt wird. Setzt man in die Funktion
```
Y = 3 * X + 5
```
das Argument 2 ein, so nimmt die Funktion und damit der Funktionswert den Wert 11 an (6 plus 5).
Auflösung: Maß für die Feinheit einer Grafik, eines Bildes, das aus vielen kleinen Punkten (*Pixel*) zusammengesetzt ist. Die Auflösung gibt an, wie viele Pixel das Bild horizontal und vertikal enthält, die Auflösung eines Hires-Bildes auf dem C64 ist beispielsweise 320 x 200.

Augpunkt: siehe Perspektive
Bogenmaß: Maß für die Größe eines Winkels; im Gegensatz zum Gradmaß geht die Skala beim Bogenmaß von 0 bis 2 mal Pi (entspricht 360°). Formel:
$Winkel_{(Bogenmaß)} = Winkel_{(Gradmaß)}$ mal Pi / 180°

Bei den trigonometrischen Funktionen des C64 werden Winkel im Bogenmaß angegeben. Einige Werte:

Bogenmaß	Dezimalwert	Gradmaß
Pi	3,14159	180°
Pi/2	1,57080	90°
Pi/4	0,78540	45°
2 Pi	6,28319	360°
0	0	0°

Cosinus: Trigonometrische Funktion; siehe Bild 10: Der Term
```
XS = R * COS(W)
```
liefert die Koordinate XS des Punktes P auf dem Kreis mit dem Radius R, wenn der Winkel W beträgt (W im Bogen- oder Gradmaß). Siehe auch Sinus.
Definitionsbereich oder **Definitionsmenge**: Zahlenmenge; enthält alle Werte, die man in eine Funktion einsetzen darf. Die Funktion
```
Y = 4 * X
```
hat als Definitionsmenge sämtliche reellen Zahlen, es gibt keine Einschränkungen. Die Funktion
```
Y = SQR (X)
```
also Y = Wurzel aus X hat als Definitionsbereich die Menge aller Zahlen, die größer oder gleich Null sind; nur diese dürfen für X eingesetzt werden.
Dimension: allg. Ausmaß, Ausdehnung, Bereich; kleinste Anzahl von Koordinaten, mit denen ein Objekt beschrieben werden kann; ein Punkt hat die Dimension Null, eine Linie die Dimension 1, eine ebene Fläche (Ebene) die Dimension 2, der gewöhnliche Raum (räumlich) hat drei Dimensionen. Nimmt man z. B. noch die *Zeit* dazu, kommt man auf den vierdimensionalen Raum.
Drehung (*Rotation*): die Bewegung eines Objektes um einen festen Punkt (*Zentrum*) oder eine feste Achse (hier: eine der drei Koordinatenachsen); im Gegensatz zur Verschiebung (*Translation*), bei der sich alle Punkte des Objekts auf parallelen und gleich langen Bahnen bewegen.
Durchmesser: doppelter Radius
Einheit: Bezugsgröße in einem Koordinatensystem. Vereinbart man etwa, die Lage der X-Koordinate »1« soll 3 cm vom Nullpunkt des Koordinatensystems entfernt liegen, ist die Einheit 3 cm.
Ellipse: flachgedrückter Kreis, eiförmig; Kreis mit zwei unterschiedlichen Radien; Darstellung mit Funktionstermen:
```
X = R1 * cos (W)
Y = R2 * sin (W)
```
Siehe auch Bild 10, Sinus und Cosinus.
Euler, Leonhard; * Basel 15. April 1707, + Petersburg 18. September 1783; bedeutender schweizer. Mathematiker; Eulersche *Koordinaten* sind Drehungs-Koordinaten.
Funktion: Zuordnungsvorschrift, die gewissen Zahlen X (das »Argument«) wieder neue Zahlen Y (»Funktionswert«) zuordnet. Jedem X wird *genau* ein Y zugeordnet. Schreibweise:
```
Y = f(X)
```
Beispiele für Funktionen:
```
Y = 4 * X + 5
Y = SIN(X)
Y = PEEK(X)
Y = 0
```
Funktionsgraph: bildliche Darstellung einer Funktion (meist) in einem zweidimensionalen Koordinatensystem. Dabei werden auf der Abszisse die Argumente der Funktion angetragen, auf der Ordinate die dazugehörigen Funktionswerte.
Funktionsterm: Term in der Funktion, Algorithmus. Bei der Funktion
```
Y = 4 * X + 3
```
lautet der Term
```
4 * X + 3
```
Funktionswert: Ergebnis einer Funktion. Setzt man in die Funktion
```
Y = 3 * X + 5
```
das Argument 2 ein, so nimmt die Funktion und damit der Funktionswert den Wert 11 an (6 plus 5).
Gerade: mathematischer Ausdruck für eine unendlich lange Linie. Eine Gerade mit einer besonderen Aufgabe wird *Achse* genannt, das Teilstück heißt *Strecke*.
Gradmaß: Maß für die Größe eines Winkels; im Gegensatz zum Bogenmaß geht die Skala beim Gradmaß von 0 bis 360 Grad (entspricht 2 mal Pi). Formel:
$Winkel_{(Gradmaß)} = Winkel_{(Bogenmaß)}$ mal 180° / Pi
Siehe dazu auch Bogenmaß
Graph siehe Funktionsgraph
horizontal: waagerecht
Interpolation (oder **Approximation**): angenäherte Bestimmung oder Darstellung einer Größe oder Funktion. So lassen sich z. B. komplizierte Funktionen durch einfachere ausdrücken, ohne daß der Fehler einen ganz bestimmten Wert übersteigt. Durch Approximation kann z. B. ein Funktionswert ungefähr bestimmt werden, angenähert, approximiert werden, weil seine exakte Berechnung einen sehr großen Rechenaufwand erfordern würde, oder weil er zwischen zwei bekannten Meßwerten liegt; man spricht dann von *Interpolation*.

Speziell beim Drucker bedeutet I., daß künstlich die Auflösung gesteigert wird, indem auch in den Halbzeilen zwischen zwei Zeilen auf dem Papier Punkte gesetzt werden, wenn dies z. B. eine Fläche schwärzer macht; siehe dazu 64'er Magazin, Ausgabe 1/89, Seite 122.
Kartesische Koordinaten: Art der Darstellung eines Punktes: In Bild 10 sind die Streckenlängen XS und YS die kartesischen Koordinaten PX/PY des Punktes P. In einem kartesischen Koordinatensystem stehen die beiden Achsen senkrecht aufeinander, und es gilt überall dieselbe Einheit (Beispiel »1« entspricht 1 cm).
komplex: läßt sich nicht in unserem Zahlensystem darstellen, etwa die

wird eingeschaltet, es geht mit dem Befehl !DRAW weiter. Wegen des speziellen Formats können Grafiken zwischen Prograf und anderen Programmen nicht direkt ausgetauscht werden.

Hardcopy: !COPY,Typ (,Sekundäradresse)

Natürlich darf einem solchen Top-Programm auch eine gute Druckfunktion nicht fehlen. Diese Anweisung erzeugt auf einem Epson-Drucker eine Hardcopy der Grafik. Die Sekundäradresse für das Interface muß so gewählt werden, daß dieses die Daten nicht wandelt (Einstellung »Direktkanal«, siehe Handbuch zum Interface).

Ganz nach dem Motto »für jeden Geschmack etwas« bietet die Hardcopy-Funktion vier verschiedene Arten der Hardcopy:

Typ=0 **(schnell)**: Ausdruck der Grafik in 640 x 200 Punkten in der Standard-Dichte. Je zwei übereinanderliegende Punkte werden ODER-verknüpft und ergeben einen gedruckten Punkt.

Typ=1 **(klein)**: Die 640 x 400 Punkte der Grafik werden in der größtmöglichen Punktdichte gedruckt, also waagrecht mit 1920 dpi, senkrecht im Abstand $1/216$ Zoll.

Typ=2 **(mittel)**: Der Punktabstand ist gegenüber Typ 1 waagerecht und senkrecht verdoppelt, die dazwischenliegenden Punkte werden durch Interpolation ermittelt.

Typ=3 **(groß)**: Der Punktabstand ist gegenüber Typ 1 verdreifacht, die dazwischenliegenden 3 x 3 Punkte werden durch Interpolation ermittelt.

Zubehör serienmäßig

Zusammen mit den interpolierten Punkten ist die Punktdichte bei den Typen 1 bis 3 also maximal. Der Zeitaufwand für den Ausdruck ist beträchtlich, da für eine Bildschirmzeile sechs Durchläufe des Druckkopfs nötig sind. Der Druckvorgang kann durch Abschalten des Druckers oder durch

Lexikon

Wurzel aus einer negativen Zahl; man definiert die Variable »I«:

`i = Wurzel aus -1`

So kann jede komplexe Zahl in der Form

`Zahl = A + i * B`

dargestellt werden.

konstant: gleichbleibend, nicht veränderlich, nicht variabel

Koordinate: Lagebezeichnung eines Punktes in einem Koordinatensystem. Beispiel New York: Ein Fußgänger steht an der Kreuzung 5. Straße West, 17. Straße Nord. Dann lauten seine Koordinaten bezüglich des New Yorker Koordinatensystems 5 und 17 oder (andere Darstellung) (5/17).

Koordinatensystem: System, in dem die Lage von jedem Punkt in einem Raum (zwei-, drei-, ... dimensional) in Form von Zahlen ausgedrückt werden kann. Üblich ist das Kartesische K., bei dem die beiden Koordinatenachsen senkrecht aufeinander stehen und dieselben Einheiten haben.

negativ: kleiner als Null; Vorzeichen Minus; Gegenteil von positiv

Objekt: irgendein Gegenstand

optional: wahlfrei, muß nicht unbedingt verwendet werden

Ordinate: zweite Koordinate eines Punktes, gewöhnlich die (senkrechte) Y-Achse. Siehe auch Abszisse

Parameter: Hilfsvariable; Parameter in einer Funktionsvorschrift legen das Aussehen des Graphen näher fest; hinter einem Basic-Befehl sind Parameter nähere Angaben, z. B. der Filename nach dem LOAD-Befehl.

Parameterdarstellung: mathematische Beschreibung einer Kurve, Fläche oder eines Funktionsgraphen durch Angabe der Koordinaten aller Punkte, die in der Kurve/Fläche/Graph vorkommen als Funktionen einer oder mehrerer unabhängiger veränderlicher Größen, der sog. Parameter. Beispiel:

`X = X(t), Y = Y(t), Z = Z(t)`

(Parameter = t), oder

```
X = X(W) = R * cos (W)
Y = Y(W) = R * sin (W)
```

(ergibt einen Kreis, Parameter ist der Winkel W; der Radius R ist kein Parameter, da er konstant bleibt. Siehe auch Sinus und Cosinus sowie Bild 10.)

Parameterkurve: Kurve, Funktion in der Parameterdarstellung

Perspektive: Die zweidimensionale, bildliche Darstellung dreidimensionaler Objekte mit Hilfe einer Zentralprojektion (*Zentralperspektive*) oder mit Hilfe einer Parallelprojektion (Parallelperspektive), die dem Betrachter ein anschauliches, naturgetreues Bild vermitteln soll; die Zentralperspektive ist dem natürlichen einäugigen Sehprozeß nachgebildet; man denkt sich ein Auge, das bestimmte Koordinaten (*Augpunkt*) im Raum hat, und durch eine große Glasscheibe hindurch einen räumlichen Gegenstand beobachtet. Das Auge (ein einziger Punkt!) sendet gedachte »Sehstrahlen« zum Objekt (wie dünne Schnüre) aus; wo diese die Glasplatte durchstoßen, wird auf der Platte ein Punkt gezeichnet. Am Ende erhält man auf der Platte die Zentralperspektive des betrachteten Objekts dahinter. Wie man sieht, hängt die Perspektive von mehreren Faktoren ab: Die Koordinaten des Auges, die Lage der Platte, die Koordinaten des Objektes. Siehe auch Projektion.

Pi: π, irrationale mathematische Konstante mit dem Wert 3,141 592 653 589 793 238 462 643 383 279 502 884 ... Wird vor allem in der Kreisberechnung eingesetzt: Kreisfläche = Radius^2 mal Pi; Pi dient auch zur Winkelumrechnung vom/ins Bogenmaß: $\text{Pi}^\circ_{\text{Bogenmaß}} = 180^\circ_{\text{Bogenmaß}}$

Pixel: Einzelpunkt in einer Grafik, siehe Auflösung

Polarkoordinaten: Art der Darstellung eines Punktes. Siehe Bild 10: Die Polarkoordinaten des Punktes P auf dem Kreis mit dem Mittelpunkt M und Radius R sind R und der Winkel W. Gegenteil: kartesische Koordinaten.

positiv: größer als Null; Vorzeichen Plus; Gegenteil von negativ

Projektion: Abbildung, bei der die Bildfigur (Abbildung) dadurch entsteht, daß man von den Punkten des Originalgebildes (Objektes) Geraden zeichnet, die entweder parallel sind (Parallelprojektion) oder alle durch einen bestimmten Augpunkt (Zentrum, Zentralperspektive) gehen und diese Gerade mit der Projektionsebene (vgl. Glasplatte bei Perspektive) zum Schnitt bringt. Die Schnittpunkte sind die Bildpunkte, der sogenannte *Riß*. Siehe auch Perspektive.

Quadrant: Teilbereich eines Koordinatensystems; das übliche Koordinatensystem mit den beiden Achsen X und Y ist in vier Quadranten unterteilt, die durch diese beiden Achsen getrennt und rechts oben (X und Y positiv) beginnend entgegen dem Uhrzeigersinn mit römischen Zahlen durchnumeriert sind.

Radius: charakteristische Größe des Kreises, gibt seine Ausmaße an: Die Entfernung vom Mittelpunkt zu einem beliebigen Kreispunkt ist der Radius (in Bild 10 mit R bezeichnet).

reell: läßt sich in unserem Zahlensystem darstellen, Gegenteil von komplex. Reelle Zahlen lassen sich durch ganze Zahlen (z. B. 15), durch periodische Dezimalbrüche (etwa $1/3$) oder nichtperiodische Dezimalbrüche (etwa 15.456, pi oder Wurzel aus 2) darstellen.

reelle Zahl: siehe reell

reelle Funktion: eine Funktion (Abbildung), deren Funktionswerte reelle Zahlen sind

Riß: siehe *Projektion*

Rotation: siehe *Drehung*

Schrägbild: Betrachtet man einen Körper, der in einem dreidimensionalen Raum steht, aus einer bestimmten Perspektive, so erhält man das S. Beispiel: Das S. des Kreises ist eine Ellipse.

Sinus: Trigonometrische Funktion; siehe Bild 10: Der Term

`YS = R * SIN(W)`

liefert die Höhe YS des Punktes P auf dem Kreis mit dem Radius R, wenn der Winkel W beträgt (W im Bogen- oder Gradmaß). Siehe auch Cosinus.

Strecke: Teilstück einer Geraden

Term: Rechenausdruck, Formel

Translation: (Parallel-)Verschiebung eines Objektes (bezügl. einer Achse), geradlinig fortschreitende Bewegung eines Körpers; Bild 9 zeigt eine Translation bezüglich der X-Achse.

trigonometrische Funktion: z. B. Sinus, Cosinus, Tangens; Sammelbegriff für Winkelfunktionen, Kreisfunktionen; Umkehrfunktionen: »zyklometrische Funktionen«

Ursprung: Nullpunkt eines Koordinatensystems. Hier treffen sich die beiden Koordinatenachsen, alle Koordinaten haben den Wert Null.

Winkel: Grad für die Neigung zweier Geraden, die sich schneiden, aber nicht in dieselbe Richtung weisen

variabel: veränderlich, frei wählbar; Gegenteil von konstant

vertikal: senkrecht

Zentralperspektive: siehe *Perspektive*

Zentrum: allg. Mitte, Mittelpunkt; siehe auch *Drehung*.

Tastendruck (Taste drücken, nicht loslassen und etwas warten!) abgebrochen werden.

Damit wären alle Befehle und Funktionen der Grafikerweiterung »Prograf« erläutert. Der eine oder andere von Ihnen ist jetzt von dieser Vielfalt vielleicht »erschlagen«. Damit Sie erst einmal ein bißchen mit den Funktionen »spielen« und sich mit den vielen Eigenarten anfreunden können, wurde das Basic-Programm »Funprint« (Listing 4, bitte bei abgeschalteter Erweiterung mit dem Checksummer auf Seite 159 eingeben) entwickelt. Es nutzt die Fähigkeiten von »Prograf« speziell für das Zeichnen von Funktionsgraphen (fast) aller Funktionen, deren Graph sich zwei- oder dreidimensional darstellen läßt. Darüber hinaus eignet es sich ganz hervorragend zur einfachen Konstruktion von Schrägbildern in Zentralperspektive.

Folgende Arten von Funktionen lassen sich mit »Funprint« verarbeiten:

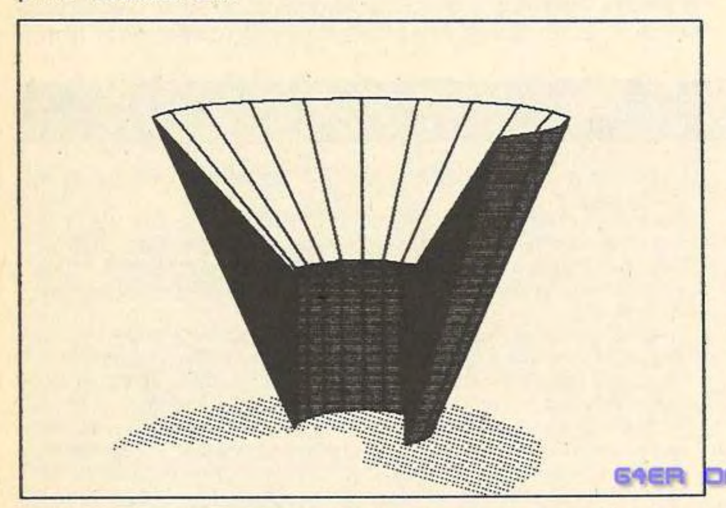


Bild 7. Davon könnte auch M.C. Escher noch etwas lernen!

Bild	Parameter
Nr. 4	X = I Y = J Z = SIN(I-J+SIN(I+J))/(3+J/3) I von —26 bis 14, Schritt 0.5 J von —4 bis 30, Schritt 0.6 X = I Y = 0 Z = LOG(3+I*I)+3 I von —7 bis 7, Schritt 0.2 X = I Y = 0 Z = LOG(1+I*I)+3 I von —7 bis 7, Schritt 0.2 Ursprung: 320/340, Einheit: 40 Proj.zent.: 180/—600/300 keine Rotation, keine Translation
Nr. 5	X = COS(I)*(4+SIN(I*3.3)) Y = SIN(I)*(3+SIN(I*3.3)) I von 0 bis 63, Schritt 0.1 Ursprung (320/200), Einheit 40
Nr. 6	X = I Y = J Z = -2/SQR(I*I+J*J+.1) I von —3 bis 3 Schritt 0.5 J von —3 bis 3 Schritt 0.5 X = I*COS(J) Y = I*SIN(J) Z = —2/I I von 0.5 bis 3, Schritt 0.3 J von 0 bis 6.29, Schritt 0.314 Ursprung (320/60), Einheit 55 Proj.Zentrum 180/—1100/300 Keine Drehung, Translation: —3/0/—2

Tabelle 4. Die Parameter einiger Bilder

1. Reelle Funktionen mit einer reellen Variablen, z.B.
   `f(x) = x+sin(x)`
2. 2D-Parameterkurven, wie
   `x(i) = i.sin(i)`
   `y(i) = i.cos(i)`
   (ergibt eine Spirale)
3. 3D-Parameterkurven, etwa
   `x(i) = sin(i)`
   `y(i) = cos(i)`
   `z(i) = i`
   (das Ergebnis ist eine Schraubenlinie)
4. zwei- oder dreidimensionale Strecken mit Koordinaten
5. zweiparametrige Flächen und alle Funktionen, die sich als solche darstellen lassen, wie reelle Funktionen zweier Variablen oder reelle Funktionen einer komplexen Variablen. Beispiel:
   `x(i,j) = sin(i)cos(j)`
   `y(i,j) = sin(i)sin(j)`
   `z(i,j) = cos(i)`
   Diese Vorschrift ergibt eine Kugel. Oder:
   `f(z) = abs(2*z*Pi) mit z = x+iy`

Großer Wert wurde auf die Benutzerfreundlichkeit gelegt: Eigentlich müßten vor dem Zeichnen von Graphen etliche Parameter definiert werden (man denke an Definitionsbereich, Lage von Ursprung und Projektionszentrum, Einheit). Dies alles wird vom Programm jedoch soweit wie möglich automatisiert, so daß nur wenige Tasten gedrückt werden müssen, bevor die Grafik entsteht. Feste Voreinstellungen und die anderen Optionen, die »Funprint« bietet, sind zusammen mit den benötigten Tasten übersichtlich auf dem Bildschirm dargestellt (Bild 2 zeigt das Einschaltbild dieses Programms). Der Name »Funprint« leitet sich übrigens von »**Fun**ction **Print**er« ab und hat nur wenig mit dem englischen Wort »fun« zu tun. Obwohl die Arbeit mit dem Programm nun wirklich viel Spaß bereiten kann!

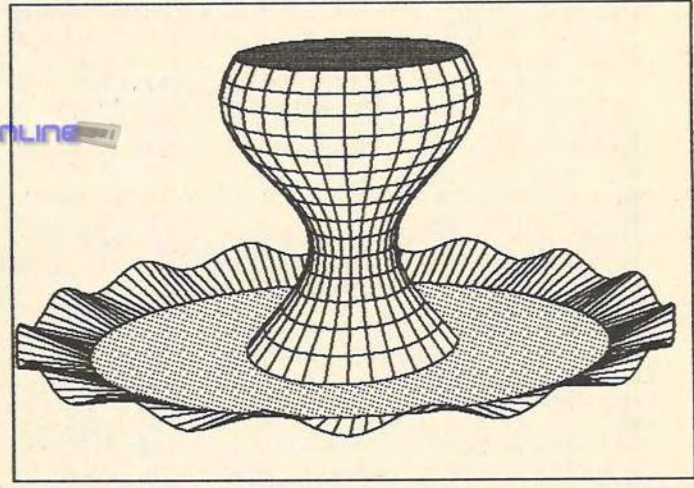


Bild 8. Der Kelch wartet auf seine Füllung

Eine gewisse Aufmerksamkeit des Benutzers muß allerdings vorausgesetzt werden, wenn Programmabstürze vermieden werden sollen: Die Funktionsterme müssen in korrekter Basic-Syntax geschrieben werden, und die Parameter dürfen nur erlaubte Bereiche durchlaufen. So führt die Eingabe von

```
X(I)=SQR(I)
```

als Funktion mit I im Bereich von —1 bis 1 unweigerlich zum Ausstieg: Wurzeln aus negativen Zahlen gibt es im reellen Bereich nicht. In einem solchen Fall gelangt man durch Eingabe von

```
GOTO 10
```

ohne Verlust der Variablen wieder ins Programm. Da die Funktionen nicht punktweise mit !PLOT, sondern Schritt für

Schritt mit !LINE gezeichnet werden, beeinflußt die Schrittweite die Genauigkeit, aber auch die Geschwindigkeit beim Zeichnen. Große oder genau berechnete Schrittweiten liefern auch gewollte Effekte: So kann aus einer Ellipse ohne weiteres ein Vieleck oder eine Rosette werden, wie dies in Bild 3 eindrucksvoll gezeigt wird. Die Parameter für dieses Bild:

```
X = 5*SIN(I)
Y = 3*COS(I)
I von 0 bis 6.3
Ursprung 320/200
Einheit 45(/50/55)
Schrittweite:
a) 0,1   b) 0,786   c) 1,698 (Endwert von I bei c)
nicht 6.3, sondern 63)
```

Hier erkennt man klar die Formel für eine typische Ellipse (siehe Textkasten Mathelexikon).

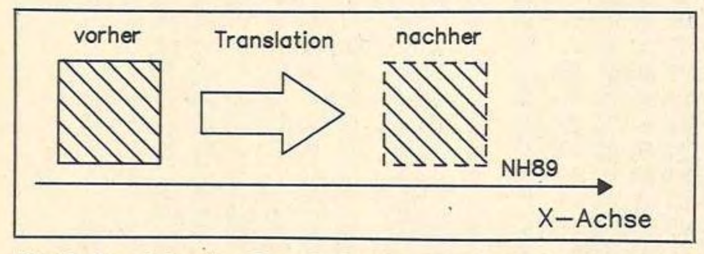


Bild 9. So sieht eine Translation bezüglich der X-Achse aus

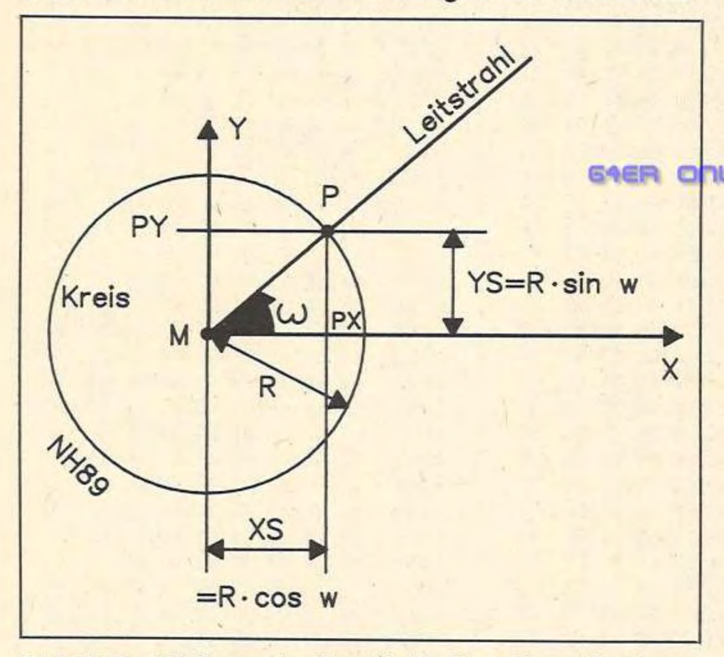


Bild 10. Zur Erklärung der Begriffe Radius, Sinus, Cosinus und Polarkoordinaten (Mathelexikon)

Das Programm zeichnet beliebig viele Funktionen nacheinander in die gleiche Grafik und merkt sich bis zu 35 Funktionsterme. Hat man etwa bereits eine etwas aufwendigere Grafik generiert, möchte aber gern den Ursprung, das Projektionszentrum oder die Einheit verändern, so kann man nach dem Löschen des Bildes mit der Taste <F8> die ganze Grafik neu entstehen lassen. <F7> zeichnet mit Gedächtnis, <F5> ohne Gedächtnis, <F8> zeichnet alle gemerkten Terme. Der Zeichenvorgang kann jederzeit mit <RETURN> abgebrochen oder mit <←> zum Bearbeiten mit dem Joystick unterbrochen werden (siehe dazu die Beschreibung von »Prograf«). Speziell die »verdeckten« Linien in 3D-Graphen lassen sich durch Anklicken eines Punktes der Linie »elegant« löschen. Einen eigenen Hidden-Line-Algorithmus, der das automatisch erledigt, enthält das Programm nicht.

Natürlich fehlt auch nicht die Möglichkeit, die erzeugten Grafiken auf Diskette zu speichern oder mit einem Epson-Drucker zu Papier zu bringen. In den Zeilen 450 und 460 können die Sekundäradressen für das Interface verändert werden.

Tasten	Funktion
<F1>	wählt die Darstellungsform, zur Auswahl stehen: Funktion, Kurve 2D, Kurve 3D, Fläche, Strecke 2D und Strecke 3D
<F3>	wählt die Termnummer zum Zeichnen, falls bereits mit <F7> mehrere Terme gespeichert wurden. <F3> alleine zählt aufwärts, mit SHIFT wird abwärts gezählt
<F4>	siehe <F3>
<F5>	Zeichnen ohne Gedächtnis
<F7>	Zeichnen mit Gedächtnis, kann mit <F8> wieder abgerufen werden
<F8>	gesamte Grafik noch einmal aufbauen (Nummer wird mit <F3> gewählt, die gewünschten Grafiken werden mit <F7> fixiert)
<L>	LOAD, Grafik von Diskette laden: Namen eingeben, RETURN-Taste drücken (Leereingabe bricht ab)
<S>	SAVE, Grafik auf Diskette speichern: Namen eingeben, RETURN-Taste drücken (Leereingabe bricht ab)
<D>	Druck, druckt Grafik auf angepaßten Drucker aus. Sie können im Druckmenü mit folgenden Tasten Sonderfunktionen wählen: <T>: Text ein/aus; <R>: Rand ein/aus; <0> bis <3>: Druckmodus wie beschrieben wählen; <RETURN> startet den Druck
<+>,<->, <£>,<Klammeraffe>, <*>,<)>, <:>,<;> und <=>	Vorwahlen für Koordinatensystem-Ursprung (so kann man wählen, welche(r) Quadrant(en) vornehmlich gezeichnet werden sollen
<K>	Ursprung zeigen; die Koordinaten des Ursprungs können, wenn sie angezeigt werden, mit den Tasten <X>, <SHIFT X>, <Y> und <SHIFT Y> verändert werden; <RETURN> beendet diese Funktion
<CLR>	löschen: <B> Bild, <T> Term oder <G> alles
<RETURN>	Grafik zeigen und Editiermodus wie beschrieben starten
<U>	wählt X-Komponente der Rotation
<V>	wählt Y-Komponente der Rotation
<W>	wählt Z-Komponente der Rotation
<M>	wählt X-Komponente der Translation
<N>	wählt Y-Komponente der Translation
<O>	wählt Z-Komponente der Translation
<0>	schaltet Koordinatensystem in der Grafik ein und aus
<1> bis <5>	Vorwahlen für Projektionszentrum
<6>	zeigt Koordinaten des Projektionszentrum, können mit den Tasten <X>, <Y> und <Z> mit oder ohne <SHIFT> geändert werden, Ende dieser Funktion mit <RETURN>
<CRSR>	mit den Cursortasten wird der Maßstab geändert
Die folgenden Funktionen arbeiten nur, wenn die entsprechende Funktion mit der F1-Taste gewählt ist:	
<A>	legt Koordinaten von Punkt A fest: Die Koordinaten werden nacheinander eingegeben und mit <RETURN> fixiert
<B>	legt Koordinaten von Punkt B fest
<F>	Funktion eingeben (Text in Basic-Syntax <!> eingeben und mit <RETURN> speichern
<I>	wählt »für I von ... bis ... Schrittweite ...«
<J>	dito, für Variable J
<X>	legt X(I) usw. fest
<Y>	legt Y(I) usw. fest
<Z>	legt Z(I) usw. fest

Tabelle 5. Die Bedienung von »Funprint« (Listing 4). Anmerkung der Redaktion: Da diese Tabelle nicht vom Programmautor stammt, ist es nicht ganz ausgeschlossen, daß sie in ganz kleinen Details nicht stimmt oder es noch weitere Funktionen gibt.

Wir haben für Sie herausgefunden, was das Programm »Funprint« alles kann, und in der Tabelle 5 alle Tasten und Funktionen dieses Programms übersichtlich zusammengestellt.

Falls Sie lieber sofort etwas »Konkretes« auf dem Bildschirm sehen wollen, geben Sie das Demoprogramm (Listing 5) mit dem Checksummer ein und starten es mit RUN. Das Programm zeichnet dann einige Demobilder auf den Schirm. Der Aufbau vor allem des zweiten Bildes dauert etwas, gedulden Sie sich bitte. Nach Abschluß der Zeichnung verfärbt sich der Bildschirmrahmen schwarz, dann drücken Sie eine Taste, um das nächste Bild zu sehen.

Jetzt liegt es bei Ihnen, was Sie mit diesen Spitzenprogrammen anfangen. Zum Appetitanregen und Neugierigmachen dienen die Bilder 4 bis 8, die noch einmal deutlich machen, was alles möglich ist. Die verwendeten Parameter (für »Funprint«) entnehmen Sie bitte der Tabelle 4.

(Bertram Hafner/Nikolaus Heusler/ef)

```
Name : prograf                  4360 5c00
------------------------------------------
4360 : 50 4c 4f d4 4c 49 4e c5  19
4368 : 5a 4c 49 4e c5 5a 50 4c  0d
4370 : 4f d4 44 52 41 d7 9c 53  71
4378 : 43 ce 4c 4f c3 50 52 4f  c6
4380 : da 52 4f d4 54 52 41 4e  6b
4388 : d3 96 43 4f 50 d9 94 93  ae
4390 : 00 00 00 00 58 48 8b 4e  23
4398 : f9 50 e5 50 40 53 e0 46  eb
43a0 : 44 47 a3 47 06 50 11 45  0b
43a8 : d3 44 20 51 a9 4b ba 59  62
43b0 : 88 45 00 00 00 a0 92 a2  70
43b8 : 00 a5 7a 85 fd a5 7b 85  e0
43c0 : fe c8 c8 a5 fd 85 7a a5  4a
43c8 : fe 85 7b 20 73 00 29 7f  47
43d0 : 85 02 bd 60 43 85 fb d0  c4
43d8 : 05 a2 0b 4c 37 a4 29 7f  b7
43e0 : e8 4c fa 43 20 73 00 c9  47
43e8 : 21 f0 06 20 79 00 4c e7  a0
43f0 : a7 20 b5 43 4c ae a7 ea  2c
43f8 : ea ea c5 02 f0 0c a5 fb  07
4400 : 29 80 d0 bd bd 60 43 e8  13
4408 : d0 f6 a5 fb 10 bd 8c 15  87
4410 : 44 20 73 00 6c 94 43 00  ba
4418 : 20 25 48 10 03 4c cb 53  47
4420 : 20 39 56 d0 f8 a2 01 8e  52
4428 : 12 46 ca 8e be 4a 86 c6  c8
4430 : 20 35 57 ba 8e bb 4a a5  53
4438 : 02 c9 05 90 03 4c e5 57  4b
4440 : 4c 4a 57 a9 01 8d aa 44  6c
4448 : a5 15 a4 14 20 53 44 a5  1c
4450 : af a4 ae 20 87 44 ad 05  5c
4458 : 01 20 69 44 ad 06 01 20  9c
4460 : 69 44 ad 07 01 20 69 44  77
4468 : 60 20 91 55 ac aa 44 ee  f6
4470 : aa 44 b9 a3 44 a8 a2 00  33
4478 : bd b3 4a 99 00 5a c8 c8  5c
4480 : c8 e8 e0 07 d0 f2 60 20  3c
4488 : 95 b3 20 dd bd b9 01 01  6a
4490 : f0 03 c8 d0 f8 a2 00 b9  66
4498 : fe 00 9d 05 01 c8 e8 e0  5a
44a0 : 03 d0 f4 60 42 5a c2 ea  2c
44a8 : eb ec 00 20 55 45 a9 f6  22
44b0 : a0 44 20 67 b8 a2 b3 20  17
44b8 : 49 45 20 59 45 a9 fb a0  aa
44c0 : 44 20 67 b8 a2 b8 20 49  08
44c8 : 45 20 4e 45 a9 f1 a0 44  8f
44d0 : 4c 67 b8 20 00 50 a2 f1  f3
44d8 : a0 44 20 d4 bb 20 00 50  9a
44e0 : a2 f6 a0 44 20 d4 bb 20  86
44e8 : 00 50 a2 fb a0 44 4c d4  3f
44f0 : bb 00 00 00 00 00 00 00  ac
44f8 : 00 00 00 00 00 00 00 00  f9
4500 : 20 00 50 a2 76 a0 46 4c  a7
4508 : d4 bb a9 76 a0 46 4c a2  a6
4510 : bb a9 25 20 c9 50 a9 00  b3
4518 : 8d 43 47 20 00 45 f0 03  11
4520 : 8d 43 47 20 6b e2 20 37  e1
4528 : 45 20 0a 45 20 64 e2 20  9a
4530 : 37 45 c6 fe d0 e5 60 a6  a6
4538 : fd a0 47 20 d4 bb 18 a5  32
4540 : fd 69 05 85 fd 60 00 a2  0c
4548 : a9 a0 4a 4c d4 bb a9 ae  8d
4550 : a0 4a 4c a2 bb a9 b3 d0  f6
4558 : f7 a9 b8 d0 f3 a5 fd a0  12
4560 : 47 4c 28 ba 20 86 49 a9  dd
4568 : 76 a0 46 20 50 b8 a2 80  1a
4570 : 20 05 45 a9 71 20 0c 45  6c
4578 : a9 80 a0 46 20 0f bb 20  fc
4580 : bf b1 a5 65 a4 64 60 60  de
4588 : ce ac 59 20 a2 59 ee ac  11
4590 : 59 a5 90 d0 65 20 c6 ff  6c
4598 : a0 00 84 fd a9 80 85 fe  cc
45a0 : a2 08 86 b1 20 e4 ff 85  52
45a8 : fb a9 00 26 fb 90 03 20  cd
45b0 : e4 ff 78 a2 30 86 01 a0  83
45b8 : 00 91 fd 20 b3 48 a5 90  39
45c0 : d0 10 e6 fd c6 b1 d0 e3  17
45c8 : e6 fd c6 fd d0 d2 e6 fe  5b
45d0 : 30 ce 20 14 4c a2 01 20  10
45d8 : 4a 47 a9 00 a8 85 fb a9  2a
45e0 : e0 85 fc 20 a0 4b 20 b9  1e
45e8 : 48 a5 fc 18 69 80 85 fc  f0
45f0 : c8 d0 f3 e6 fc d0 ef 20  50
45f8 : b3 48 4c 40 53 ea ea ea  f8
4600 : 00 00 00 00 00 00 00 00  01
4608 : 00 00 00 00 00 0b 00 00  61
4610 : 00 02 ff 60 80 a0 [illegible] e0  ef
4618 : 00 c0 fe 00 00 40 01 80  3f
4620 : 02 c0 03 00 05 40 06 80  af
4628 : 07 c0 08 00 0a 40 0b 80  61
4630 : 0c c0 0d 00 0f 40 10 80  14
4638 : 11 c0 12 00 14 40 15 80  c7
4640 : 16 c0 17 00 19 40 1a 80  79
4648 : 1b c0 1c 00 1e 01 02 04  c4
4650 : 08 10 20 40 80 40 01 c8  10
4658 : 00 02 00 7c 02 02 00 8c  32
4660 : 01 00 ff 04 00 88 70 00  e8
4668 : 00 8b 89 80 00 88 48 00  06
4670 : 00 00 00 00 00 00 00 00  71
4678 : 00 00 00 00 00 00 00 00  79
4680 : 00 00 00 00 00 86 fb 84  ae
4688 : fc a2 00 8e 10 46 8e 11  37
4690 : 46 ca 8e 12 46 ea a6 fb  6f
4698 : f0 28 a9 2c a0 00 d1 7a  d2
46a0 : d0 3d 20 73 00 20 8a ad  0c
46a8 : 20 bf b1 ee 10 46 ad 10  fc
46b0 : 46 18 2a aa a5 65 9d ff  de
46b8 : 45 a5 64 9d 00 46 c6 fb  e2
46c0 : d0 d8 a6 fc f0 19 a9 2c  1c
46c8 : a0 00 d1 7a d0 11 20 9b  79
46d0 : b7 8a ee 11 46 ae 11 46  55
46d8 : 9d 0c 46 c6 fc d0 e7 60  9c
46e0 : a2 00 a0 02 20 85 46 ae  8f
46e8 : 11 46 f0 21 ad 0e 46 18  11
46f0 : 2a 2a 2a 2a 0d 0d 46 a0  93
46f8 : e8 a2 5b 84 fb 86 fc a2  c6
4700 : 04 a0 18 91 fb c8 d0 fb  ce
4708 : e6 fc ca d0 f6 20 a0 4b  c2
4710 : a0 00 84 fb a9 60 85 fc  fe
4718 : 98 91 fb c8 d0 fb e6 fc  13
4720 : d0 f7 4c b3 48 00 49 0f  3d
4728 : da a2 80 7f ff ff ff 00  63
4730 : 49 0f da a2 80 7f ff ff  10
4738 : ff 00 49 0f da a2 80 7f  2f
4740 : ff ff ff 00 20 fd ae 20  2c
4748 : 9e b7 a9 03 0d 02 dd 8d  00
4750 : 02 dd ad 00 dd e0 00 f0  73
4758 : 1c 29 fc 09 02 8d 00 dd  b1
4760 : a9 0f 2d 18 d0 09 78 8d  32
4768 : 18 d0 a9 20 0d 11 d0 a2  39
4770 : 97 a0 47 d0 18 09 03 8d  34
4778 : 00 dd ad 18 d0 29 01 09  42
4780 : 14 8d 18 d0 ad 11 d0 29  74
4788 : df a2 8b a0 e3 8d 11 d0  40
4790 : 8e 00 03 8c 01 03 60 86  27
4798 : 02 a2 00 20 4a 47 a6 02  6d
47a0 : 6c 00 03 a2 06 a0 00 20  c7
47a8 : 85 46 a2 04 bd 00 46 9d  aa
47b0 : 54 46 ca d0 f7 ae 10 46  b6
47b8 : e0 06 d0 47 38 ad 05 46  4a
47c0 : e9 80 ad 06 46 30 3c e9  c0
47c8 : 02 10 38 38 ad 07 46 e9  e7
47d0 : 80 ad 08 46 30 2d e9 02  0a
47d8 : 10 29 38 ad 09 46 e9 90  cc
47e0 : ad 0a 46 30 1e e9 01 10  7f
47e8 : 1a 38 ad 0b 46 e9 90 ad  3c
47f0 : 0c 46 30 0f e9 01 10 0b  0a
47f8 : a2 08 bd 04 46 9d 58 46  cd
4800 : ca d0 f7 60 18 6d 55 46  0b
4808 : 85 ae 98 6d 56 46 85 af  c5
4810 : 60 85 14 84 15 38 ad 57  41
4818 : 46 e5 14 85 14 ad 58 46  a3
4820 : e5 15 85 15 60 38 a5 ae  50
4828 : ed 59 46 a5 af ed 5a 46  68
4830 : 30 25 38 ad 5b 46 e5 ae  93
4838 : ad 5c 46 e5 af 30 18 38  af
4840 : a5 14 ed 5d 46 a5 15 ed  d8
4848 : 5e 46 30 0b 38 ad 5f 46  32
4850 : e5 14 ad 60 46 e5 15 60  5f
4858 : a2 02 a0 01 20 85 46 ae  e8
4860 : 11 46 f0 06 ad 0d 46 8d  09
4868 : 12 46 a2 01 20 78 4e ea  3b
4870 : ea 20 25 48 10 01 60 20  88
4878 : ce 48 7d 14 46 85 fc a5  1c
4880 : aa 29 07 49 07 aa bd 4d  01
4888 : 46 78 a0 30 84 01 a0 00  0b
4890 : ae 12 46 38 30 0c f0 04  0f
4898 : 51 fb b0 08 49 ff 31 fb  65
48a0 : b0 02 11 fb 91 fb ad b2  2a
48a8 : 4a 18 69 e0 85 fc a0 00  37
48b0 : 20 b9 48 a9 37 85 01 58  48
48b8 : 60 a2 04 a9 00 11 fb 48  a9
48c0 : 38 a5 fc e9 20 85 fc 68  3a
48c8 : ca d0 f2 91 fb 60 a5 ae  a0
48d0 : 85 aa a5 af 85 ab a5 14  7e
48d8 : 85 ac a5 15 85 ad a9 00  2c
48e0 : 18 66 ad 66 ac 2a 66 ab  70
48e8 : 66 aa 2a aa a5 ac 4a 4a  01
48f0 : 4a 0a a8 a5 ac 29 07 18  7e
48f8 : 79 1b 46 85 fb b9 1c 46  cc
4900 : 85 fc a5 aa 29 f8 65 fb  aa
4908 : 85 fb a5 ab 65 fc 85 fc  b8
4910 : 8d b2 4a 18 60 ea ea a2  da
4918 : 05 a9 1b 20 d2 ff a9 33  f7
4920 : 20 d2 ff 8a 20 d2 ff a9  e6
4928 : 0a 20 d2 ff 60 ea a2 01  e1
4930 : d0 e7 a2 0d d0 e3 a2 15  1f
```

```
4938 : d0 df a2 18 d0 db a9 1b  6c
4940 : 20 d2 ff a9 40 d0 dd a2  46
4948 : 03 a9 1b 20 d2 ff a9 2a  13
4950 : 20 d2 ff 8a 4c d2 ff a2  cb
4958 : 01 20 49 49 a9 80 20 d2  aa
4960 : ff a9 02 d0 ef 20 47 49  7e
4968 : 4c 5c 49 20 47 49 a9 00  9e
4970 : 20 d2 ff a9 05 d0 dd 20  bd
4978 : 47 49 a9 80 20 d2 ff a9  ca
4980 : 07 4c d2 ff a9 69 a0 46  57
4988 : 4c 28 ba 20 a0 4b ea ea  81
4990 : a0 00 a9 04 8d be 4a ae  71
4998 : bd 4a ca d0 02 a0 04 a2  c1
49a0 : 00 b1 fb 85 aa b1 b3 85  3a
49a8 : ab b1 b1 85 ac b1 b5 26  c4
49b0 : aa 3e a9 4a 26 ab 3e a9  39
49b8 : 4a 26 ac 3e aa 4a 2a 3e  2a
49c0 : aa 4a e8 e8 e0 10 d0 e7  88
49c8 : c8 ce be 4a d0 d1 b1 b1  b6
49d0 : 85 ae b1 fb c0 08 d0 1a  5c
49d8 : 18 a5 fb 69 38 85 aa 85  54
49e0 : ac a5 fc 69 01 85 ab 18  e6
49e8 : 69 20 85 ad b1 ac 85 ae  6c
49f0 : b1 aa a2 08 2a 2e bb 4a  38
49f8 : 2e ba 4a 26 ae 2e bb 4a  bb
4a00 : 2e ba 4a ca d0 ee ae bd  32
4a08 : 4a ca f0 07 a0 0c b1 fb  fd
4a10 : 38 b0 04 a0 10 b1 aa 8d  0a
4a18 : bc 4a 18 a5 fb 69 08 85  ea
4a20 : fb 85 b1 85 b3 85 b5 d0  db
4a28 : 08 e6 fc e6 b2 e6 b4 e6  c2
4a30 : b6 a0 00 ae bd 4a ca d0  07
4a38 : 02 a0 04 a2 04 b1 fb 2a  f2
4a40 : 2e b9 4a b1 b3 2a 2e b9  cc
4a48 : 4a c8 ca d0 f0 ac bf 4a  cb
4a50 : 88 d0 06 8e b9 4a 8e bc  35
4a58 : 4a ac c0 4a 88 d0 0f ac  16
4a60 : bd 4a 88 d0 09 8e ba 4a  03
4a68 : 8e bb 4a 8e bc 4a 4c b3  ef
4a70 : 48 ea ea ea ad c1 4a 26  a4
4a78 : fd 2a 26 fe 2a c0 01 f0  82
4a80 : 0e 26 fd 2a 26 fe 2a aa  be
4a88 : 6a 66 fe 6a 66 fd 60 2e  66
4a90 : bc 4a 2e bb 4a 2e ba 4a  0a
4a98 : 2a 2e bb 4a 2e ba 4a 2a  48
4aa0 : aa 6a 6e ba 4a 6e bb 4a  0e
4aa8 : 60 ff ff ff ff ff ff ff  08
4ab0 : ff ff ff ff ff ff ff ff  af
4ab8 : ff ff ff ff ff ff ff ff  b7
4ac0 : 00 00 00 00 00 00 00 00  c1
4ac8 : 00 00 00 00 00 00 d0 00  0c
4ad0 : 00 00 00 00 00 00 00 00  d1
4ad8 : 00 00 80 80 00 00 80 80  0c
4ae0 : 00 00 00 00 00 00 00 00  e1
4ae8 : 00 00 80 80 00 00 80 80  1c
4af0 : 00 00 00 00 00 00 00 00  f1
4af8 : 80 80 80 80 80 80 80 80  f8
4b00 : 00 00 00 00 00 00 80 80  04
4b08 : 00 00 00 80 00 00 80 80  1c
4b10 : 00 00 00 00 00 00 00 00  11
4b18 : 00 80 00 80 00 80 80 80  70
4b20 : 00 00 00 00 00 00 00 00  21
4b28 : 00 00 00 00 80 80 80 80  38
4b30 : 00 00 00 00 00 00 00 80  32
4b38 : 00 80 00 80 00 80 00 80  8e
4b40 : 00 00 00 00 00 00 80 80  44
4b48 : 00 00 00 00 00 80 80 80  50
4b50 : 00 00 00 00 00 00 00 00  51
4b58 : 00 00 00 00 80 80 80 80  68
4b60 : 00 00 00 00 00 00 00 00  61
4b68 : 00 00 80 80 00 00 80 80  9c
4b70 : 00 00 00 00 00 00 00 00  71
4b78 : 80 80 80 80 80 80 80 80  78
4b80 : 00 00 00 00 00 00 80 80  84
4b88 : 00 80 00 80 00 80 80 80  e0
4b90 : 00 00 00 00 00 00 00 00  91
4b98 : 00 00 00 00 80 80 80 80  a8
4ba0 : 78 a9 30 85 01 60 ea ea  3e
4ba8 : ea a2 00 a0 02 20 85 46  bb
4bb0 : a0 ff ad 11 46 c9 02 d0  3a
4bb8 : 03 ac 0e 46 a2 04 20 af  88
4bc0 : 59 20 c9 ff a9 00 85 c6  da
4bc8 : 85 fb 85 b1 85 b3 85 b5  5a
4bd0 : a9 80 85 fc a9 a0 85 b2  d5
4bd8 : a9 c0 85 b4 a9 e0 85 b6  fe
4be0 : 20 3e 49 a9 19 8d c0 4a  3c
4be8 : ae 0d 46 d0 03 4c 1d 4c  68
4bf0 : ca d0 03 4c f4 4d ca d0  f3
4bf8 : 03 4c 36 4d 4c bf 4c 20  8d
4c00 : b7 ff 30 0c a5 c6 d0 05  22
4c08 : ce c0 4a d0 db 20 3e 49  2d
4c10 : a9 00 85 c6 20 cc ff a9  af
4c18 : 20 20 c3 ff 60 a9 28 8d  48
4c20 : bf 4a 20 57 49 a0 00 20  d1
4c28 : a0 4b ea ea a2 00 b1 fb  6f
4c30 : 11 b3 2a 3e a9 4a e8 e8  d0
4c38 : e0 10 d0 f6 a2 00 b1 b1  88
4c40 : 11 b5 2a 3e aa 4a e8 e8  f1
4c48 : e0 10 d0 f6 c8 c0 08 d0  98
4c50 : db 20 b3 48 ea ea 18 a5  e3
4c58 : fb 69 08 85 fb 85 b1 85  78
4c60 : b3 85 b5 d0 08 e6 fc e6  d7
4c68 : b2 e6 b4 e6 b6 a2 00 bd  93
4c70 : a9 4a 20 d2 ff e8 e0 10  8c
4c78 : d0 f5 ce bf 4a d0 a6 20  f4
4c80 : 3a 49 4c ff 4b a5 fb 8d  5f
4c88 : c2 4a a5 fc 8d c3 4a a5  e4
4c90 : b2 8d c4 4a a5 b4 8d c5  45
4c98 : 4a a5 b6 8d c6 4a 60 ad  b0
4ca0 : c2 4a 85 fb 85 b1 85 b3  cb
4ca8 : 85 b5 ad c3 4a [illegible] fc ad  0c
4cb0 : c4 4a 85 b2 ad c5 4a 85  8e
4cb8 : b4 ad c6 4a 85 b6 60 a9  21
4cc0 : 02 8d bd 4a 20 85 4c a9  f4
4cc8 : 03 8d c1 4a 20 9f 4c 20  bc
4cd0 : 77 49 a9 28 8d bf 4a 20  9b
4cd8 : 8b 49 a9 00 8d c7 4a ae  10
4ce0 : c7 4a bd a9 4a 85 fd bd  b5
4ce8 : aa 4a 85 fe a0 08 20 74  ac
4cf0 : 4a bd c0 4a 2a 2e c8 4a  5e
4cf8 : bd f0 4a 2a 2e c9 4a bd  db
4d00 : 20 4b 2a 2e ca 4a 88 d0  d9
4d08 : e5 b9 c8 4a 20 d2 ff c8  6f
4d10 : c0 03 d0 f5 ee c7 4a ad  f6
4d18 : c7 4a c9 10 d0 c1 ce bf  4e
4d20 : 4a d0 b4 20 2e 49 ce c1  ef
4d28 : 4a d0 a1 20 36 49 ce bd  ab
4d30 : 4a d0 94 4c ff 4b a9 02  96
4d38 : 8d bd 4a 20 85 4c a9 03  a2
4d40 : 8d c1 4a 20 9f 4c 20 6b  f8
4d48 : 49 a9 28 8d bf 4a 20 8b  07
4d50 : 49 a9 00 8d c7 4a ae c7  39
4d58 : 4a bd a9 4a 85 fd bd aa  c9
4d60 : 4a 85 fe a0 08 8c cb 4a  e9
4d68 : a9 0d 18 6d c1 4a 8d ca  86
4d70 : 4a c9 0e d0 04 26 fd 26  f2
4d78 : fe ad ca 4a 4a a8 a9 02  dd
4d80 : 90 03 a9 01 c8 20 77 4a  1c
4d88 : bd 50 4b 2a 2e c8 4a bd  53
4d90 : 70 4b 2a 2e c9 4a ce cb  b8
4d98 : 4a ad ca 4a c9 04 90 12  d8
4da0 : 29 01 f0 04 26 fd 26 fe  ef
4da8 : a0 03 ce ca 4a 88 d0 fa  f9
4db0 : f0 c7 0e c8 4a 0e c9 4a  f1
4db8 : ce cb 4a d0 f5 ad c8 4a  9d
4dc0 : 20 d2 ff ad c9 4a 20 d2  14
4dc8 : ff ee c7 4a ad c7 4a c9  4f
4dd0 : 10 d0 83 ce bf 4a f0 03  1b
4dd8 : 4c 4e 4d 20 2e 49 ce c1  8f
4de0 : 4a f0 03 4c 43 4d 20 32  70
4de8 : 49 ce bd 4a f0 03 4c 3e  26
4df0 : 4d 4c ff 4b a9 02 8d bd  29
4df8 : 4a 20 85 4c a9 03 8d c1  aa
4e00 : 4a 20 9f 4c 20 65 49 a9  71
4e08 : 28 8d bf 4a 20 8b 49 a9  07
4e10 : 00 8d c7 4a ae c7 4a bd  e0
4e18 : a9 4a 85 fd a9 03 38 ed  77
4e20 : c1 4a aa f0 05 26 fd ca  de
4e28 : d0 fb a9 05 18 6d c1 4a  89
4e30 : a8 a9 08 8d cb 4a 26 fd  04
4e38 : 26 fe ce cb 4a c0 04 30  25
4e40 : 09 26 fd 26 fd 88 88 88  f8
4e48 : d0 ec 06 fe ce cb 4a d0  06
4e50 : f9 a5 fe 20 d2 ff ee c7  58
4e58 : 4a ae c7 4a e0 10 d0 b7  76
4e60 : ce bf 4a d0 a7 20 2e 49  81
4e68 : ce c1 4a d0 94 20 17 49  fd
4e70 : ce bd 4a d0 87 4c ff 4b  3b
4e78 : bd 00 46 bc 01 46 20 04  29
4e80 : 48 bd 02 46 bc 03 46 20  2d
4e88 : 11 48 60 a2 00 86 b1 86  32
4e90 : b2 86 02 a2 04 a0 01 20  e4
4e98 : 85 46 ae 11 46 f0 06 ad  6d
4ea0 : 0d 46 8d 12 46 a2 01 20  34
4ea8 : 78 4e a5 ae a4 af 85 b3  cc
4eb0 : 84 b4 a5 14 a4 15 85 b5  ef
4eb8 : 84 b6 20 25 48 10 04 a2  9e
4ec0 : 01 86 02 a2 05 20 78 4e  a9
4ec8 : 20 25 48 10 04 a2 01 86  f5
4ed0 : 02 38 a5 b3 e5 ae 8d 61  9b
4ed8 : 46 a5 b4 e5 af 8d 62 46  58
4ee0 : 10 23 a5 b3 a4 b4 a6 ae  49
4ee8 : 86 b3 a6 af 86 b4 85 ae  69
4ef0 : 84 af a5 b5 a4 b6 a6 14  2f
4ef8 : 86 b5 a6 15 86 b6 85 14  01
4f00 : 84 15 38 b0 cc 38 a5 b5  c3
4f08 : e5 14 8d 63 46 a5 b6 e5  ff
4f10 : 15 8d 64 46 10 13 a2 01  f4
4f18 : 86 b2 38 a5 14 e5 b5 8d  1c
4f20 : 63 46 a5 15 e5 b6 8d 64  c5
4f28 : 46 38 ad 63 46 ed 61 46  48
4f30 : ad 64 46 ed 62 46 30 1c  b0
4f38 : ad 61 46 ac 62 46 ae 63  97
4f40 : 46 8e 61 46 ae 64 46 8e  33
4f48 : 62 46 8d 63 46 8c 64 46  84
4f50 : a2 01 86 b1 18 ad 61 46  4c
4f58 : 2a 85 b3 ad 62 46 2a 85  f3
4f60 : b4 18 ad 63 46 2a 85 b5  2f
4f68 : ad 64 46 2a 85 b6 38 a5  58
4f70 : b5 ed 61 46 85 fd a5 b6  89
4f78 : ed 62 46 85 fe ee 61 46  52
4f80 : d0 04 ea ee 62 46 a6 02  e2
4f88 : f0 06 20 71 48 4c 93 4f  85
4f90 : 20 77 48 a5 fe 30 2f a4  aa
4f98 : b1 d0 0d a4 b2 d0 12 e6  51
4fa0 : 14 d0 16 e6 15 4c b9 4f  b8
4fa8 : e6 ae d0 0d e6 af 4c b9  4c
4fb0 : 4f a5 14 d0 02 c6 15 c6  29
4fb8 : 14 38 a5 fd e5 b3 85 fd  1f
4fc0 : a5 fe e5 b4 85 fe a4 b1  3a
4fc8 : f0 0d a4 b2 d0 12 e6 14  20
4fd0 : d0 16 e6 15 4c e8 4f e6  1f
4fd8 : ae d0 0d e6 af 4c e8 4f  ae
4fe0 : a5 14 d0 02 c6 15 c6 14  5c
4fe8 : a5 fd 18 65 b5 85 fd a5  09
4ff0 : fe 65 b6 85 fe ce 61 46  78
4ff8 : d0 8c ce 62 46 10 87 60  d2
```

Listing 1. Bitte geben Sie die Grafikerweiterung »Prograf« mit dem MSE (Seite 159) ein

```
5000 : 20 fd ae 4c 8a ad 20 00   eb
5008 : 50 a2 65 20 05 45 20 00   02
5010 : 50 a2 69 20 05 45 20 00   0b
5018 : 50 a2 6d 4c 05 45 20 c7   29
5020 : 50 20 00 50 a6 fd 20 49   f8
5028 : 45 20 3e 45 c6 fe d0 f1   41
5030 : ad 43 47 f0 5c a9 2a 20   6b
5038 : c9 50 20 4e 45 20 5d 45   51
5040 : 20 03 45 20 55 45 38 a5   e3
5048 : fd e9 05 48 20 5f 45 a9   ea
5050 : 76 a0 46 20 50 b8 20 47   86
5058 : 45 20 4e 45 68 20 5f 45   79
5060 : 20 03 45 20 55 45 20 5d   12
5068 : 45 a9 76 a0 46 20 67 b8   a8
5070 : a2 ae 20 49 45 20 59 45   e0
5078 : a2 b3 20 49 45 a9 a9 20   ae
5080 : 50 45 a2 b8 20 49 45 20   d4
5088 : 3e 45 20 3e 45 c6 fe d0   61
5090 : a9 20 ab 44 20 87 51 a9   94
5098 : 69 a0 46 20 50 b8 a5 61   0b
50a0 : d0 07 a9 e8 a0 bf 20 a2   49
50a8 : bb a2 71 20 05 45 20 55   ba
50b0 : 45 a9 65 20 64 45 20 04   20
50b8 : 48 20 59 45 20 87 51 a9   e6
50c0 : 6d 20 64 45 4c 11 48 a9   c1
50c8 : ae 85 fd a9 03 85 fe 60   06
50d0 : ea ea ea ea a2 ff a0 00   f4
50d8 : a9 2c d1 7a d0 03 20 9b   38
50e0 : b7 8e 12 46 60 20 1e 50   4c
50e8 : 20 d4 50 4c 71 48 20 1e   26
50f0 : 50 20 25 48 10 02 e6 02   53
50f8 : 60 a2 00 86 b1 86 b2 86   a1
5100 : 02 20 ee 50 a5 ae a4 af   9a
5108 : 85 b3 84 b4 a5 14 a4 15   d6
5110 : 85 b5 84 b6 20 ee 50 20   63
5118 : d4 50 4c d1 4e ea ea ea   1f
5120 : 20 e1 b3 a0 00 b1 4e 85   04
5128 : fb c8 b1 4e 85 fc d0 03   47
5130 : 4c ae b3 a0 00 a9 3a 91   2e
5138 : fb c8 b1 fb d0 f7 84 fd   5e
5140 : a5 7a 48 a5 7b 48 a2 00   6e
5148 : 20 7e a5 88 88 88 88 88   22
5150 : 88 c4 fd 90 05 a2 17 4c   26
5158 : 37 a4 b9 00 02 91 fb 88   fd
5160 : 10 f8 68 85 7b 68 85 7a   bd
5168 : 60 b9 1a 46 85 fc a5 ae   28
5170 : 38 e9 08 29 f0 85 aa a5   f5
5178 : af e9 00 38 30 01 18 6a   63
5180 : 85 ab 66 aa 4c 02 49 20   04
5188 : 84 49 20 03 45 4c 55 45   b0
5190 : 50 56 30 36 50 36 30 56   02
5198 : 50 56 38 42 18 b2 20 6a   d6
51a0 : c0 ad 00 d0 29 80 8d 00   1e
51a8 : d0 a5 14 38 e9 08 29 0f   f9
51b0 : 0a 18 69 46 8d 01 d0 a5   59
51b8 : 15 6a a5 14 6a 18 69 32   60
51c0 : 8d 0f d0 a5 ae 38 e9 08   22
51c8 : 29 0f 0a 18 69 40 0d 00   cb
51d0 : d0 8d 00 d0 a5 af 18 6a   8e
51d8 : 48 a5 ae 6a 18 69 18 8d   34
51e0 : 0e d0 68 69 00 f0 07 ad   9d
51e8 : 10 d0 09 80 d0 05 ad 10   bf
51f0 : d0 29 7f 8d 10 d0 a5 ae   62
51f8 : 49 08 29 0f f0 13 c9 0f   5e
5200 : f0 0f a5 14 49 08 29 0f   fb
5208 : f0 07 c9 0f f0 03 4c 43   af
5210 : 44 a0 08 a5 af f0 0e b9   89
5218 : 00 d0 29 7f 99 00 d0 88   a9
5220 : 88 10 f4 30 0c b9 00 d0   24
5228 : 09 80 99 00 d0 88 88 10   6b
5230 : f4 ad 0f d0 38 e9 36 29   d7
5238 : f8 18 69 32 8d 0d d0 a5   ad
5240 : af 18 6a 48 a5 ae 6a 38   89
5248 : e9 04 29 f8 85 aa 68 e9   c0
5250 : 00 48 18 a5 aa 69 18 8d   a1
5258 : 0c d0 68 69 00 29 01 f0   43
5260 : 07 ad 10 d0 09 40 d0 05   3c
5268 : ad 10 d0 29 bf 8d 10 d0   c1
5270 : a0 00 a5 14 38 e9 08 29   41
5278 : f0 85 ac a5 15 e9 00 85   b6
5280 : ad 30 09 6a a5 ac 6a 4a   d3
5288 : 4a a8 c8 c8 b9 19 46 85   fa
5290 : fb ea ea 20 69 51 69 80   87
5298 : 85 fc 18 69 20 85 b2 69   9a
52a0 : 20 85 b4 69 20 85 b6 a5   31
52a8 : fb 85 b1 85 b3 85 b5 20   01
52b0 : a0 4b a9 02 8d cf 4a a9   74
52b8 : 00 85 fd a9 5a 85 fe d0   9f
52c0 : 24 a9 80 85 fd 18 a5 fb   b9
52c8 : 69 40 85 fb 85 b1 85 b3   95
52d0 : 85 b5 a5 fc 69 01 85 fc   e7
52d8 : 18 69 20 85 b2 69 20 85   60
52e0 : b4 69 20 85 b6 a0 00 c0   f4
52e8 : 08 d0 07 18 a5 fd 69 10   2d
52f0 : 85 fd b1 fb 85 aa b1 b1   37
52f8 : 85 ab 20 34 53 91 fd e6   69
5300 : fd 20 34 53 91 fd e6 fd   25
5308 : e6 fd b1 b3 85 aa b1 b5   af
5310 : 85 ab 20 34 53 91 fd e6   81
5318 : fd 20 34 53 91 fd e6 fd   3d
5320 : c8 c0 10 d0 c2 ce cf 4a   dd
5328 : d0 97 20 b3 48 4c 43 44   bf
5330 : ea ea ea ea a2 04 26 aa   df
5338 : 2a 26 ab 2a ca d0 f7 60   79
5340 : a2 00 a0 02 20 85 46 ad   ed
5348 : 0d 46 ae 11 46 d0 02 a9   8c
5350 : 0e a0 06 99 27 d0 88 d0   20
5358 : fa ad 0e 46 ae 11 46 d0   a3
5360 : 02 a9 01 8d 2e d0 8d 27   17
5368 : d0 a2 11 e0 0c d0 02 a2   7e
5370 : 0a bd 8f 51 9d ff cf ca   15
5378 : d0 f1 a9 1e 8d 1d d0 8d   8f
5380 : 17 d0 a0 6d 8c fe 5f 88   24
5388 : 8c ff 5f 8c f8 5f a2 04   9a
5390 : 88 98 9d f8 5f ca d0 f8   6c
5398 : 86 02 e8 20 4a 47 a9 c0   64
53a0 : 8d 15 d0 a9 10 85 ae 85   14
53a8 : 14 a2 02 86 af ca 86 15   f4
53b0 : a6 02 bc f6 5b a2 00 b9   a3
53b8 : 80 5b 9d 30 5b c8 e8 e0   b5
53c0 : 0c d0 f4 ea ea 20 a1 51   a8
53c8 : 20 11 52 a0 20 ca d0 fd   b1
53d0 : 88 d0 fa 84 c6 ea ea ad   da
53d8 : 00 dc 29 1f ea ea ea c9   ba
53e0 : 1f f0 27 a2 df 8e 15 d0   fe
53e8 : 4a b0 03 20 7a 54 4a b0   24
53f0 : 03 20 89 54 4a b0 03 20   67
53f8 : 9a 54 4a b0 03 20 a9 54   e5
5400 : 48 20 a1 51 68 4a 90 4b   9d
5408 : b0 cd a0 0a ca d0 fd 88   44
5410 : d0 fa a5 c6 f0 c1 ad 77   62
5418 : 02 86 c6 c9 5f f0 20 c9   da
5420 : 11 d0 02 a2 fd c9 91 d0   84
5428 : 02 a2 fe c9 9d d0 02 a2   22
5430 : fb c9 1d d0 02 a2 f7 e0   48
5438 : 00 f0 7f 8a 4c e3 53 c6   a0
5440 : 02 10 04 a2 08 86 02 4c   f5
5448 : b0 53 a2 00 8e 15 d0 20   60
5450 : 4a 47 60 a6 02 f0 0f ca   a4
5458 : d0 03 4c d9 56 ca d0 03   fd
5460 : 4c 38 59 4c 18 44 e8 8e   0d
5468 : 12 46 20 71 48 20 11 52   42
5470 : ad 00 dc 29 10 f0 f9 4c   83
5478 : cb 53 a6 14 d0 08 c6 15   ac
5480 : 10 04 a2 01 86 15 c6 14   af
5488 : 60 e6 14 d0 0c e6 15 a2   0c
5490 : 02 e4 15 d0 04 a2 00 86   c6
5498 : 15 60 a6 ae d0 08 c6 af   24
54a0 : 10 04 a2 02 86 af c6 ae   fa
54a8 : 60 e6 ae d0 0c e6 af a2   3d
54b0 : 03 e4 af d0 04 a2 00 86   8e
54b8 : af 60 c9 1c d0 08 a2 00   65
54c0 : 8e cb 4a 4c cb 53 c9 9f   0e
54c8 : d0 04 a2 ff d0 f2 c9 92   34
54d0 : d0 08 a2 00 8e cd 4a 4c   66
54d8 : cb 53 c9 12 d0 04 a2 ff   b9
54e0 : d0 f2 c9 90 d0 08 a2 d0   27
54e8 : 8e ce 4a 4c cb 53 c9 05   82
54f0 : d0 04 a2 d8 d0 f2 c9 88   63
54f8 : d0 06 20 7f 55 4c c5 53   39
5500 : c9 8c d0 06 20 76 55 4c   a8
5508 : c5 53 c9 87 d0 06 20 6d   73
5510 : 55 4c c5 53 c9 8b d0 06   af
5518 : 20 64 55 4c c5 53 c9 20   a8
5520 : d0 03 4c 17 56 8d cc 4a   01
5528 : 29 7f c9 0d d0 03 4c 4a   10
5530 : 54 c9 14 d0 03 4c 1d 56   3c
5538 : c9 20 30 af ea ad cc 4a   f7
5540 : 10 10 c9 ff d0 05 a9 5e   63
5548 : 4c 88 55 29 7f 09 40 4c   2d
5550 : 88 55 c9 40 30 f9 c9 60   b8
5558 : 10 05 29 1f 4c 88 55 29   ca
5560 : 5f 4c 88 55 a0 08 20 7a   72
5568 : 54 88 d0 fa 60 a0 08 20   ff
5570 : 89 54 88 d0 fa 60 a0 08   a5
5578 : 20 9a 54 88 d0 fa 60 a0   b3
5580 : 08 20 a9 54 88 d0 fa 60   49
5588 : 20 c0 55 20 7f 55 4c c5   c1
5590 : 53 0a 85 fb a9 00 2a 06   18
5598 : fb 2a 06 fb 2a 18 ae cd   63
55a0 : 4a f0 02 69 04 6d ce 4a   8c
55a8 : 85 fc 78 a9 33 85 01 a0   a3
55b0 : 07 b1 fb 99 b3 4a 88 10   92
55b8 : f8 a9 37 85 01 58 60 ea   2e
55c0 : a2 ff 8e 12 46 20 91 55   9e
55c8 : 20 64 55 ae cb 4a d0 03   9e
55d0 : 20 64 55 a9 00 8d c9 4a   d5
55d8 : 20 89 54 ae cb 4a d0 03   00
55e0 : 20 89 54 a9 07 8d ca 4a   ac
55e8 : ae c9 4a 3e b3 4a 90 11   c7
55f0 : 20 71 48 ae cb 4a d0 09   15
55f8 : 20 89 54 20 71 48 20 7a   c5
5600 : 54 20 a9 54 ce ca 4a 10   e6
5608 : df 20 76 55 ee c9 4a ae   03
5610 : c9 4a e0 08 d0 c2 60 20   1c
5618 : 26 56 4c 8b 55 20 76 55   c9
5620 : 20 26 56 4c c5 53 a2 00   f4
5628 : 8e 12 46 a2 07 a9 ff 9d   9e
5630 : b3 4a ca 10 fa 20 c8 55   3b
5638 : 60 20 ce 48 7d 14 46 85   02
5640 : fc a5 aa 29 07 49 07 aa   0b
5648 : bd 4d 46 8d b1 4a 78 a2   84
5650 : 30 86 01 a0 00 31 fb f0   73
5658 : 02 a9 ff a8 a2 37 86 01   44
5660 : 58 a6 02 e0 04 d0 01 c8   04
5668 : 98 60 ea ea ea a9 09 85   73
5670 : b3 a9 00 85 b1 85 b2 85   c6
5678 : b4 20 7a 54 20 c0 56 f0   a9
5680 : 3e 20 9a 54 20 c0 56 f0   43
5688 : 36 20 89 54 20 c0 56 f0   fe
5690 : 2e 20 89 54 20 c0 56 f0   fe
5698 : 26 20 a9 54 20 c0 56 f0   06
56a0 : 1e 20 a9 54 20 c0 56 f0   06
56a8 : 16 20 7a 54 20 c0 56 f0   3b
56b0 : 0e 20 7a 54 20 c0 56 f0   3b
56b8 : 06 20 9a 54 20 89 54 60   60
56c0 : e6 b4 20 25 48 30 0f 20   30
56c8 : 39 56 f0 0a a5 b1 d0 04   dd
56d0 : a5 b4 85 b1 e6 b2 c6 b3   ed
56d8 : 60 20 25 48 10 03 4c cb   7d
56e0 : 53 20 39 56 f0 f8 a2 00   be
56e8 : 8e 12 46 20 77 48 20 6d   2a
```

```
56f0 : 56 a6 b2 e0 02 d0 26 20   e1
56f8 : 35 57 a9 01 8d c1 4a a5   bf
5700 : b1 20 6f 56 20 77 48 20   87
5708 : 6d 56 a6 b2 ca f0 f0 ce   36
5710 : c1 4a d0 06 20 20 57 4c   e4
5718 : 07 57 20 20 57 4c c5 53   6d
5720 : ad a9 4a 85 ae ad aa 4a   7d
5728 : 85 af ad ab 4a 85 14 ad   e2
5730 : ac 4a 85 15 60 a5 ae 8d   0e
5738 : a9 4a a5 af 8d aa 4a a5   08
5740 : 14 8d ab 4a a5 15 8d ac   e2
5748 : 4a 60 a9 01 8d bd 4a 8d   58
5750 : bf 4a 20 7a 54 20 25 48   f7
5758 : 30 05 20 39 56 f0 f3 20   37
5760 : 89 54 20 25 48 30 63 20   94
5768 : 39 56 d0 5e ad b1 4a 20   9e
5770 : 89 48 ba e0 20 90 e8 20   52
5778 : 9a 54 20 39 56 d0 18 ad   13
5780 : bf 4a f0 18 a5 ae 48 a5   e0
5788 : af 48 a5 14 48 a5 15 48   de
5790 : ee be 4a a9 00 f0 02 a9   88
5798 : 01 8d bf 4a 20 a9 54 20   7a
57a0 : a9 54 20 39 56 d0 18 ad   4a
57a8 : bd 4a f0 18 a5 ae 48 a5   06
57b0 : af 48 a5 14 48 a5 15 48   06
57b8 : ee be 4a a9 00 f0 02 a9   b0
57c0 : 01 8d bd 4a 20 9a 54 4c   01
57c8 : 5f 57 ce be 4a 30 13 a5   1c
57d0 : c6 d0 0f 68 85 15 68 85   7d
57d8 : 14 68 85 af 68 85 ae 4c   7e
57e0 : 4a 57 4c b0 58 20 6d 56   e8
57e8 : a5 b1 f0 03 4c cb 53 a9   c6
57f0 : 01 8d bd 4a a9 01 8d b5   b5
57f8 : 4a 8d bf 4a 20 7a 54 20   a9
5800 : 25 48 30 05 20 39 56 f0   fd
5808 : f3 20 89 54 20 25 48 30   a5
5810 : 71 20 39 56 d0 6c 20 d7   4b
5818 : 58 ba e0 20 90 eb ae bd   a8
5820 : 4a f0 06 20 a9 54 4c 2c   2f
5828 : 58 20 9a 54 20 39 56 d0   88
5830 : 3a ad bf 4a f0 3b a5 ae   57
5838 : cd b3 4a d0 15 18 a5 14   5c
5840 : ed b6 4a cd b4 4a 8d b4   12
5848 : 4a d0 07 a9 00 8d b5 4a   c9
5850 : f0 1f a5 ae 48 8d b3 4a   63
5858 : a5 af 48 a5 14 48 8d b6   c3
5860 : 4a a5 15 48 ee be 4a a9   2d
5868 : 00 f0 03 ad b5 4a 8d bf   ba
5870 : 4a ae bd 4a f0 06 20 9a   bf
5878 : 54 4c 09 58 20 a9 54 4c   79
5880 : 09 58 ce be 4a 30 16 a5   0b
5888 : c6 d0 25 68 85 15 68 85   ba
5890 : 14 68 85 af 68 85 ae ad   f8
5898 : b5 4a 4c f9 57 20 20 57   6a
58a0 : ce bd 4a 30 0b a2 00 8e   c8
58a8 : be 4a 20 9a 54 4c f4 57   11
58b0 : a9 00 85 c6 ae bb 4a 9a   bb
58b8 : 20 20 57 4c c5 53 ea ea   c0
58c0 : ea ea ea ea ea ea ea ea   bf
58c8 : ea ea ea ea ea ea ea ea   c7
58d0 : ad b1 4a 4c 89 48 ea a6   46
58d8 : 02 e0 08 d0 0a 18 a5 ae   bc
58e0 : 65 14 29 07 f0 ea 60 e0   24
58e8 : 05 d0 17 a5 14 29 01 d0   00
58f0 : 10 a6 ae a5 14 29 02 f0   28
58f8 : 01 e8 8a 45 14 29 01 f0   29
5900 : cf 60 e0 07 d0 17 a5 14   9d
5908 : 4a 4a 18 65 ae 29 03 d0   0c
5910 : 0b a5 ae 4a 4a 38 e5 14   09
5918 : 29 03 f0 b4 60 a5 14 29   6b
5920 : 01 d0 14 a5 14 29 02 d0   77
5928 : 08 a5 ae 29 03 d0 08 f0   8c
5930 : 9f a5 ae 29 01 d0 99 60   30
5938 : 20 25 48 10 03 4c cb 53   67
5940 : ae 33 5b f0 0e a5 14 cd   77
5948 : ab 4a d0 23 a5 ae cd a9   0b
5950 : 4a d0 1c a9 c0 8d 33 5b   3b
5958 : 8d 39 5b 20 35 57 20 39   5e
5960 : 56 aa f0 02 a2 01 ca 8e   02
5968 : ae 4a a2 01 4c 67 54 ae   b3
5970 : ae 4a 8e 12 46 a2 00 86   b0
5978 : b1 86 b2 86 02 a2 04 bd   aa
5980 : a8 4a 95 b2 ca d0 f8 20   60
5988 : 35 57 20 d1 4e 20 20 57   c0
5990 : a9 02 85 02 a9 00 8d 39   1f
5998 : 5b 8d 33 5b 4c 6d 54 ea   49
59a0 : ea ea 20 9e ad 20 a3 b6   b3
59a8 : 20 bd ff a0 01 a2 08 a9   53
59b0 : 20 20 ba ff 20 c0 ff a2   dc
59b8 : 20 60 20 a2 59 a5 90 d0   0b
59c0 : 3c 20 c9 ff a0 00 84 fd   97
59c8 : a9 80 85 fe 20 [illegible] 4b a2   6c
59d0 : 08 38 b1 fd d0 01 18 26   e2
59d8 : fb 9d a8 4a c8 ca d0 f1   1f
59e0 : 20 b3 48 a5 fb 20 d2 ff   ad
59e8 : a2 08 bd a8 4a f0 03 20   8b
59f0 : d2 ff ca d0 f5 c8 88 d0   f8
59f8 : d3 e6 fe d0 cf 4c 14 4c   60
5a00 : 00 00 00 00 00 00 00 00   01
5a08 : 00 00 00 00 00 00 00 00   09
5a10 : 00 00 00 00 00 00 00 00   11
5a18 : 00 00 00 00 00 00 00 00   19
5a20 : 00 00 00 00 00 00 00 00   21
5a28 : 00 00 00 00 00 00 00 00   29
5a30 : 00 00 00 00 00 00 00 00   31
5a38 : 00 00 00 00 00 00 00 00   39
5a40 : 00 00 00 00 00 00 00 00   41
5a48 : 00 00 00 00 00 00 00 00   49
5a50 : 00 00 00 00 00 00 00 00   51
5a58 : 00 00 00 00 00 00 00 00   59
5a60 : 00 00 00 00 00 00 00 00   61
5a68 : 00 00 00 00 00 00 00 00   69
5a70 : 00 00 00 00 00 00 00 00   71
5a78 : 00 00 00 00 00 00 00 00   79
5a80 : 00 00 00 00 00 00 00 00   81
5a88 : 00 00 00 00 00 00 00 00   89
5a90 : 00 00 00 00 00 00 00 00   91
5a98 : 00 00 00 00 00 00 00 00   99
5aa0 : 00 00 00 00 00 00 00 00   a1
5aa8 : 00 00 00 00 00 00 00 00   a9
5ab0 : 00 00 00 00 00 00 00 00   b1
5ab8 : 00 00 00 00 00 00 00 00   b9
5ac0 : 00 00 00 00 00 00 00 00   c1
5ac8 : 00 00 00 00 00 00 00 00   c9
5ad0 : 00 00 00 00 00 00 00 00   d1
5ad8 : 00 00 00 00 00 00 00 00   d9
5ae0 : 00 00 00 00 00 00 00 00   e1
5ae8 : 00 00 00 00 00 00 00 00   e9
5af0 : 00 00 00 00 00 00 00 00   f1
5af8 : 00 00 00 00 00 00 00 00   f9
5b00 : 00 00 00 7e 00 00 7c 00   c2
5b08 : 00 78 00 00 7c 00 00 6e   e9
5b10 : 00 00 47 00 00 03 00 00   fa
5b18 : 00 00 00 00 00 00 00 00   19
5b20 : 00 00 00 00 00 00 00 00   21
5b28 : 00 00 00 00 00 00 00 00   29
5b30 : 00 00 00 00 00 00 00 00   31
5b38 : 00 00 00 00 00 00 00 00   39
5b40 : ff ff 00 80 01 00 80 01   63
5b48 : 00 80 01 00 80 01 00 80   da
5b50 : 01 00 80 01 00 80 01 00   9a
5b58 : 80 01 00 80 01 00 80 01   7d
5b60 : 00 80 01 00 80 01 00 80   f2
5b68 : 01 00 80 01 00 ff ff 00   aa
5b70 : 00 00 00 00 00 00 00 00   71
5b78 : 00 00 00 00 00 00 00 00   79
5b80 : 00 00 00 00 00 00 00 60   41
5b88 : 00 00 00 00 0c 00 00 06   55
5b90 : 00 00 ff fc 00 01 80 00   3a
5b98 : 00 00 00 00 00 00 ff fc   92
5ba0 : 00 00 00 00 7f ff 00 7f   97
5ba8 : ff 00 7f ff 00 7f ff 00   83
5bb0 : 03 fe 00 03 fe 00 7f ff   81
5bb8 : c0 03 fe 00 cc cc 00 33   53
5bc0 : 30 00 cc cc 00 33 30 00   17
5bc8 : c0 c0 c0 0c 0c 00 00 00   5b
5bd0 : 00 c0 c0 c0 c3 0c 00 00   15
5bd8 : 00 00 00 00 00 c3 0c 00   27
5be0 : 18 61 80 30 c3 00 61 86   9e
5be8 : 00 00 00 00 00 00 00 00   e9
5bf0 : 00 00 00 00 00 00 00 0c   09
5bf8 : 18 24 30 3c 48 54 60 6c   37
```

Listing 1. (Schluß)

```
1 REM        PG.PRINTER                          <188>
2 REM -----------------------------------        <153>
3 REM   PROGRAMM ZUR DRUCKERANPASSUNG            <106>
4 REM   VON 'PROGRAF'                            <230>
5 REM -----------------------------------        <156>
10 POKE 56,67:CLR                                <148>
11 PRINT"{CLR}":PRINT" DRUCKERANPASSUNG 'P
   ROGRAF'"                                      <228>
12 PRINT"{DOWN,SPACE}SIND DIE ZEILEN 30-38
    KONTROLLIERT?"                               <199>
13 INPUT"{DOWN,SPACE}J/N";A$:IF A$<>"J"THE
   N END                                         <218>
15 SYS 57812"PROGRAF",8,1:POKE 780,0:SYS 6
   5493:REM LADEN                                <217>
19 REM ----------------------------------        <065>
20 REM KONTROLLIERE DIE FOLGENDEN CODES          <059>
21 REM ----------------------------------        <067>
30 DATA 27,64    :REM INITIALISIEREN             <231>
31 DATA 27,51    :REM ABSTAND N/216              <071>
32 DATA 27,42    :REM ESC+"*" (DICHTE)           <108>
33 DATA 3        :REM DICHTE 1920 DPL            <082>
34 DATA 1        :REM DICHTE  960 DPL            <191>
35 DATA 10       :REM 10=LINE FEED,13=CR         <060>
36 DATA 4.       :REM GERAETENR. DRUCKER         <003>
37 DATA 8        :REM GERAETENR. FLOPPY          <027>
38 DATA"NEUGRAF":REM NEUER PROGRAMMNAME          <210>
40 REM ----------------------------------        <086>
```

Listing 2. Dieses Basic-Programm nennt sich »PG.PRINTER« und dient zur Druckeranpassung. Bitte geben Sie es mit dem Checksummer (Seite 159) ein.

```
41 REM   ANPASSUNG                              <200>
50 REM ---------------------------------        <096>
60 READ A:POKE 18751,A:READ A:POKE 18756,A <238>
61 READ A:POKE 18714,A:READ A:POKE 18719,A <052>
62 READ A:POKE 18762,A:READ A:POKE 18767,A <086>
63 READ A:POKE 18760,A:READ A:POKE 18776,A <047>
64 READ A:POKE 18728,A:REM                 <127>
65 READ A:POKE 19388,A:READ A:POKE 22958,A <006>
66 PRINT"{DOWN,SPACE}DIE CODES SIND GEAEND
   ERT.{DOWN}"                             <151>
67 INPUT" BEREIT FUER 'SAVE' (J/N)";A$     <205>
68 IF A$<>"J"THEN END                      <225>
69 READ A$:PRINT"{DOWN,SPACE}SAVING";CHR$(
   34);A$;CHR$(34)                         <240>
70 OPEN 1,8,1,A$                           <247>
71 PRINT#1,CHR$(96);CHR$(67);              <040>
72 FOR I=17248 TO 23551:PRINT#1,CHR$(PEEK(
   I));:NEXT                               <052>
73 CLOSE 1:END                             <007>
```

Listing 2. (Schluß)

```
1 REM        PG.LOADER                      <036>
2 REM ---------------------------------     <153>
3 REM PROGRAMMKOPF FUER BASICPROGRAMME,     <224>
4 REM DIE 'PROGRAF' VERWENDEN.              <130>
5 REM ---------------------------------     <156>
10 POKE 52,67:POKE 56,67:CLR:REM SCHUTZ     <063>
11 IF PEEK(17803)+PEEK(22956)<>33 THEN SYS
   57812"PROGRAF",8,1:POKE 780,0:SYS 6549
   3                                        <017>
12 POKE 777,67:REM BEFEHLE AKTIVIEREN       <122>
13            :REM AUS MIT POKE777,167      <144>
```

Listing 3. »PG.LOADER« ist ein Beispiel für ein Basic-Programm zu Listing 1

```
1 POKE 649,1:POKE 56,67:CLR                 <025>
2 DEF FN A(X)=0::::::::::::::::::::::::::::
  ::::::::::::::::::::::::::::::::::::::::  <038>
3 DEF FN B(X)=0::::::::::::::::::::::::::::
  ::::::::::::::::::::::::::::::::::::::::  <055>
4 DEF FN C(X)=0::::::::::::::::::::::::::::
  ::::::::::::::::::::::::::::::::::::::::  <072>
5 X1%=0:X2%=0:Y1=0:Y2=0:X1=0:X2=0:M%=40:Z1
  =0:Z2=0:X=0:J=0:F%=0:KO%=1                <150>
6 F=55296:V=53248:GOSUB 510                 <112>
7 TX%=0:RD%=1:PZ=4:GOSUB 699:KS%=4:GOSUB 6
  49:POKE 53280,0:POKE 53281,11             <188>
8 IF PEEK(17803)+PEEK(22956)<>33 THEN SYS
  57812"PROGRAF",8,1:POKE 780,0:SYS 65493   <014>
9 POKE 777,67:!CLR,11,0:!LOC,320,200,2,636
  ,2,396                                    <236>
10 REM --- BEI FEHLER: GOTO 10 ---          <122>
11 PRINT"{GRAPHIC,CTRL-H,CLR,LIG.BLUE}";    <052>
12 PRINT"{3SPACE}U***{10SPACE}U***I         <131>
13 PRINT"{3SPACE}B{13SPACE}B{3SPACE}B{5SPA
   CE}H{6SPACE}B                            <101>
14 PRINT"{3SPACE}G* F{2SPACE}U U**I{2SPACE
   }B***K F*I{3SPACE}U**I +*                <206>
15 PRINT"{3SPACE}B{2SPACE}B{2SPACE}B B{2SP
   ACE}B{2SPACE}B{5SPACE}B{3SPACE}B B{2SPA
   CE}B B                                   <183>
16 PRINT"{3SPACE}F{2SPACE}J**K F{2SPACE}J{
   2SPACE}F{5SPACE}F{3SPACE}J F{2SPACE}J J <128>
18 REM AUTHOR B.HAFNER, 8934 GROSSAITINGEN <198>
19 REM REVISED N.HEUSLER, 8000 MUENCHEN 71 <150>
20 PRINT"{BLACK}U*********I{3SPACE}3{6SPAC
   E}U***F***I U*******I";                  <201>
21 PRINT"BFUNKTION B 2 ↑ 4{2SPACE}6UK←I B
   ↑ JIB{RVSON}+{2SPACE}-{2SPACE}£{RVOFF}B
   ";                                       <136>
22 PRINT"BKURVE{2SPACE}2DB1{2SPACE}B{2SPAC
   E}5{2SPACE}BU{GREY 2,3SPACE,BLACK}B{GRE
   Y 2,3SPACE,BLACK}MBB{RVSON,7SPACE,RVOFF
   }B";                                     <194>
23 PRINT"BFLAECHE{2SPACE}B M B N{3SPACE}BV
   {GREY 2,3SPACE,BLACK}B{GREY 2,3SPACE,BL
   ACK}NBB{RVSON}@{2SPACE}*{2SPACE}↑{RVOFF
   }B";                                     <192>
24 PRINT"BSTRECKE2DB{2SPACE}MBN <=>BW{GREY
    2,3SPACE,BLACK}B{GREY 2,3SPACE,BLACK}O
   BB{RVSON,7SPACE,RVOFF}B";                <210>
25 PRINT"JI{2SPACE}U*****K{2SPACE}*+****+*J
   ****F****KB{RVSON}:{2SPACE};{2SPACE}={R
   VOFF}B";                                 <108>
26 PRINT" BF1BU****IU****I{GREY 2,3SPACE,B
   LACK,2SPACE}TERMU***IJI{5SPACE}UK";      <058>
27 PRINT" J**KBF578BB{RVSON}L{RVOFF}DADB{4
   SPACE}F3{SPACE,GREY 2,2SPACE,BLACK}BCLR
   B B{SPACE,RVSON}K{RVOFF}S{2SPACE}B       <013>
28 PRINT" U***K{4SPACE}BB{RVSON}S{RVOFF}AV
   EJI U******K{3SPACE}JIG*****W            <048>
29 PRINT" BZEICHNENBB{RVSON}D{RVOFF}RUCKB
   B{2SPACE}LOESCHEN BBEIN{SPACE,RVSON}0{R
   VOFF}B                                   <102>
30 PRINT" J********KJ*****K J***********KJ
   *****K                                   <190>
32 FF=12:GOSUB 54:POKE FN KS(KS%),12:POKE
   F+597,12*KO%:GOSUB 801:GOSUB 802         <009>
33 POKE F+PF%(PZ),12                        <118>
35 GOSUB 781                                <187>
40 POKE 198,0:WAIT 198,1:GET D$:D=ASC(D$):
   SH=PEEK(653):IF D$<>"L"AND D$<>"S"THEN
   45                                       <198>
41 GOSUB 701:CY%=19:GOSUB 711:CX%=7:D$=NN$
   :IL=14:IF D=83 THEN 43                   <147>
42 IF D=76 THEN PRINT" LOAD:":GOSUB 721:IF
    D$<>"0"THEN NN$=D$:!LOAD D$             <047>
43 IF D=83 THEN PRINT" SAVE:":GOSUB 721:IF
    D$<>"0"THEN NN$=D$:!SAVE D$             <046>
44 GOTO 35                                  <062>
45 IF(D AND 127)=19 THEN GOSUB 771:GOTO 35 <065>
46 IF D=136 THEN GOSUB 201:GOTO 165         <181>
47 IF D=140 THEN 381                        <107>
48 IF D=135 THEN GOSUB 210:GOTO 165         <046>
49 IF D<>133 THEN 55                        <214>
50 FF=0:GOSUB 54:F%=-(F%+1)*(F%<5):FF=12:G
   OSUB 54:IF F%=0 OR F%=3 THEN 52          <192>
51 POKE 1031+FU%(F%),50-1*(F%=2 OR F%=5)    <162>
52 A$="":B$="":C$="":I1=0:I2=0:I3=0:J1=0:J
   2=0:J3=0                                 <097>
53 POKE 512,48:POKE 513,0:FL%=32:GOSUB 191
   :FL%=111:GOSUB 191:FL%=190:GOSUB 191:GO
   TO 35                                    <039>
54 FOR I=0 TO 8:POKE F+FU%(F%)+I,FF:NEXT:R
   ETURN                                    <047>
55 IF D=13 THEN 165                         <232>
56 FOR I=0 TO 9:IF D$=MID$(KS$,I+1,1)THEN
   POKE FN KS(KS%),0:KS%=I:I=11             <193>
57 NEXT:IF I=10 THEN 60                     <017>
58 POKE FN KS(KS%),12:IF KS%=9 THEN GOSUB
   611:GOTO 35                              <159>
59 GOSUB 649:GOTO 40                        <114>
60 IF D=68 THEN 401                         <182>
61 IF NOT((D=134 AND FI<FZ)OR(D=138 AND FI
   >1))THEN 64                              <075>
62 FI=FI+1+2*(D=138):FF=0:GOSUB 54:GOSUB 8
   01:GOSUB 390:GOTO 166                    <092>
64 IF D=48 THEN KO%=1-KO%:POKE F+597,12*KO
   %:GOTO 40                                <006>
65 IF D<>157 AND D<>17 THEN 70              <164>
66 IF M%>10 THEN M%=M%-5:GOSUB 801          <245>
68 IF PEEK(203)<>64 THEN 66                 <219>
69 GOTO 40                                  <015>
70 IF D<>29 THEN 74                         <020>
71 IF M%<200 THEN M%=M%+5:GOSUB 801         <099>
72 IF PEEK(203)=2 THEN 71                   <037>
73 GOTO 40                                  <019>
74 IF D<49 OR D>54 THEN 80                  <170>
75 POKE F+PF%(PZ),0:PZ=D-49:POKE F+PF%(PZ)
```

```
    ,12:IF PZ=5 THEN GOSUB 651:GOTO 35       <198>
76 GOSUB 699:GOTO 40                         <211>
80 IF(D AND 127)<85 OR(D AND 127)>87 THEN
   84                                        <244>
81 D=(D AND 3):R%(D)=R%(D)+15-30*SH:R%(D)=
   R%(D)-360*(R%(D)<0)+360*(R%(D)>350)       <123>
82 II=1286+D*40:I=R%(D):GOSUB 805:IF PEEK(
   203)<>64 THEN 81                          <192>
83 GOTO 40                                   <029>
84 IF(D AND 127)<77 OR(D AND 127)>79 THEN
   88                                        <224>
85 D=(D AND 3):T%(D)=T%(D)+1-2*SH            <207>
86 II=1286+D*40:I=T%(D):GOSUB 806:IF PEEK(
   203)<>64 THEN 85                          <212>
87 GOTO 40                                   <033>
88 ON F%+1 GOTO 91,101,121,121,141,141       <024>
90 REM --- FUNKTIONEN EINGABE ------         <106>
91 IF D$<>"F"THEN 40                         <172>
92 CX%=8:CY%=18:D$=A$:IL=66:GOSUB 721:A$=D
   $:I1=-2000:I2=2000                        <067>
93 GOSUB 181:!DEF FN A:IF KL%THEN 92         <065>
95 GOSUB 191:IF DT%THEN GOSUB 171            <074>
96 GOTO 35                                   <114>
100 REM KURVE 2D                             <140>
101 IF D$<>"X"THEN 105                       <086>
102 CY%=18:D$=A$:GOSUB 130:!DEF FN A:A$=D$
    :IF KL%THEN 102                          <138>
104 GOTO 35                                  <122>
105 IF D$<>"Y"THEN 110                       <203>
106 CY%=20:D$=B$:GOSUB 130:!DEF FN B:B$=D$
    :IF KL%THEN 106                          <094>
108 GOTO 35                                  <126>
110 IF D$<>"I"THEN 115                       <096>
111 CX%=7:CY%=16:IL=6:D$=STR$(I1):GOSUB 72
    1:I1=VAL(D$):CX%=18:D$=STR$(I2)          <068>
112 GOSUB 721:I2=VAL(D$):CX%=30:IL=8:D$=ST
    R$(I3):GOSUB 721:I3=ABS(VAL(D$))         <043>
113 IF I1>I2 THEN D=I1:I1=I2:I2=D            <169>
114 GOTO 35                                  <132>
115 IF D$<>"J"OR F%<>3 THEN 40               <022>
116 CX%=7:CY%=17:IL=6:D$=STR$(J1):GOSUB 72
    1:J1=VAL(D$):CX%=18:D$=STR$(J2)          <240>
117 GOSUB 721:J2=VAL(D$):CX%=30:IL=8:D$=ST
    R$(J3):GOSUB 721:J3=ABS(VAL(D$))         <183>
118 IF J1>J2 THEN D=J1:J1=J2:J2=D            <080>
119 GOTO 35                                  <137>
120 REM KURVE 3D /FLAECHE                    <063>
121 IF D$<>"Z"THEN 101                       <236>
122 CY%=22:D$=C$:GOSUB 130:!DEF FN C:C$=D$
    :IF KL%THEN 122                          <064>
125 GOTO 35                                  <143>
130 CX%=8:IL=66:GOSUB 721:GOSUB 181:RETURN   <033>
140 REM STRECKE                              <186>
141 IF D$<>"A"AND D$<>"B"THEN 40             <116>
142 CY%=18:IL=6:IF D$="B"THEN 150            <004>
143 CX%=1:D$=STR$(I1):GOSUB 721:I1=VAL(D$)   <226>
144 CY%=20:D$=STR$(I2):GOSUB 721:I2=VAL(D$
    ):IF F%<5 THEN 35                        <112>
145 CY%=22:D$=STR$(I3):GOSUB 721:I3=VAL(D$
    ):GOTO 35                                <193>
150 CX%=16:D$=STR$(J1):GOSUB 721:J1=VAL(D$
    )                                        <094>
151 CY%=20:D$=STR$(J2):GOSUB 721:J2=VAL(D$
    ):IF F%<5 THEN 35                        <056>
152 CY%=22:D$=STR$(J3):GOSUB 721:J3=VAL(D$
    ):GOTO 35                                <137>
165 POKE 198,0:!DRAW:GOSUB 801:GOTO 40       <017>
166 GOSUB 388:POKE F+PF%(PZ),0:PZ=5:POKE F
    N KS(KS%),0:KS%=9:GOTO 32                <159>
169 :                                        <145>
170 REM DEF.BEREICH                          <087>
171 GOSUB 701                                <195>
172 PRINT:PRINT" IN WELCHEM INTERVALL (X1,
    X2) IST":PRINT" "DT$(DT%)" IN"           <139>
173 PRINT"{GREY 2,SPACE}"D$:PRINT"{BLACK,S
    PACE}DEFINIERT?":CX%=0:CY%=23:GOSUB 71
    1                                        <189>
175 PRINT" X1 = ";TAB(20);"X2 = ":CX%=6:D$
    ="":IL=8:GOSUB 721:I1=VAL(D$)            <253>
176 CX%=25:D$="":GOSUB 721:I2=VAL(D$):RETU
    RN                                       <131>
180 REM DEF.BEREICH                          <097>
181 KL%=0:FOR I=1 TO LEN(D$):A=ASC(MID$(D$
    ,I,1)):POKE 511+I,A:IF A=40 THEN KL%=K
    L%+1                                     <101>
```

```
182 IF A=41 THEN KL%=KL%-1                   <086>
185 NEXT:POKE 511+I,0:RETURN                 <120>
191 I=0:DT%=0                                <072>
193 A=PEEK(2080+I):IF A=186 THEN DT%=DT%OR
    1                                        <074>
194 IF A=188 THEN DT%=DT%OR 2                <140>
195 IF A=174 THEN DT%=DT%OR 4                <078>
198 IF A<>58 THEN I=I+1:GOTO 193             <182>
199 RETURN                                   <001>
200 REM --- ZEICHNEN ---------------         <149>
201 IF FZ=35 THEN 210                        <218>
202 FZ=FZ+1:FI=FZ:GOSUB 205:GOTO 210         <195>
205 F%(FI)=F%:A$(FI)=A$:B$(FI)=B$:C$(FI)=C
    $:I1(FI)=I1:I2(FI)=I2:I3(FI)=I3          <251>
206 J1(FI)=J1:J2(FI)=J2:J3(FI)=J3:RETURN     <251>
210 !PROZ,P%(1),P%(2),P%(3):!LOC,OX%,OY%:!
    SCN,1                                    <041>
219 IF KO%=0 OR KR%=1 THEN 234               <101>
220 REM KOORD.SYSTEM                         <026>
222 KR%=1:!ROT,0,0,0:!TRANS,0,0,0:!LINE,-O
    X%,0,640-OX%,0:!LINE,0,OY%,0,OY%-400     <166>
225 FOR X=-M%*INT(OX%/M%)TO 640 STEP M%:!L
    INE,X,1,X,-2:NEXT                        <016>
226 FOR Y=-M%*INT(OY%/M%)TO 400 STEP M%:!L
    INE,2,-Y,-1,-Y:NEXT                      <172>
227 IF F%<2 OR F%=4 THEN 235                 <016>
228 FOR I=M%*(INT(P%(2)/M%)+4)TO 20*M%STEP
    M%:!ZPLOT,0,I,0:NEXT                     <020>
234 !ROT,R%(1)/57.3,R%(2)/57.3,R%(3)/57.3:
    !TRANS,T%(1)*M%,T%(2)*M%,T%(3)*M%        <164>
235 ON F%+1 GOTO 241,271,291,311,341,342     <205>
240 REM *** FUNKTION ***                     <166>
241 IF A$=""THEN 40                          <041>
242 X1%=-OX%:IF X1%<I1*M%THEN X1%=I1*M%      <119>
243 X3%=-OX%+639:IF X3%>I2*M%THEN X3%=I2*M
    %                                        <181>
244 IF X3%<X1%+4 THEN 40                     <165>
245 Y1=M%*FN A((X1%+.1)/M%)                  <010>
246 FOR I=X1%+.1 TO X3%-.1 STEP 4:X2%=X1%:
    Y2=Y1                                    <178>
247 X1%=I:Y1=M%*FN A(I/M%):IF ABS(Y1-Y2)<1
    20 THEN:!LINE,X1%,Y1,X2%,Y2:GOTO 250     <199>
249 IF X1%-X2%>2 THEN IF Y2>-200 THEN IF Y
    2<600 THEN I=I-2:GOTO 247                <012>
250 IF PEEK(198)THEN GOSUB 335:IF D THEN R
    ETURN                                    <144>
251 NEXT:GOTO 374                            <187>
270 REM *** KURVE 2D ***                      <206>
271 I=I1:X1=FN A(X)*M%:Y1=FN B(X)*M%         <055>
272 I=I+I3:IF I>I2 THEN 374                  <158>
273 IF PEEK(198)THEN GOSUB 335:IF D THEN R
    ETURN                                    <169>
274 X2=X1:Y2=Y1:X1=FN A(X)*M%:Y1=FN B(X)*M
    %:!LINE,X1,Y1,X2,Y2:GOTO 272             <187>
290 REM *** KURVE 3D ***                      <234>
291 I=I1:X1=FN A(X)*M%:Y1=FN B(X)*M%:Z1=FN
    C(X)*M%                                  <016>
293 I=I+I3:IF I>I2 THEN 374                  <179>
294 IF PEEK(198)THEN GOSUB 335:IF D THEN R
    ETURN                                    <190>
295 GOSUB 370:GOTO 293                       <051>
310 REM *** FLAECHE ***                      <183>
311 I=I1                                     <173>
313 J=J1:X1=FN A(X)*M%:Y1=FN B(X)*M%:Z1=FN
    C(X)*M%                                  <058>
314 J=J+J3:IF J>J2 THEN 320                  <035>
315 IF PEEK(198)THEN GOSUB 335:IF D THEN R
    ETURN                                    <211>
316 GOSUB 370:GOTO 314                       <061>
320 I=I+I3:IF I>I2 THEN 322                  <197>
321 GOTO 313                                 <123>
322 J=J1                                     <204>
323 I=I1:X1=FN A(X)*M%:Y1=FN B(X)*M%:Z1=FN
    C(X)*M%                                  <048>
324 I=I+I3:IF I>I2 THEN 330                  <198>
325 IF PEEK(198)THEN GOSUB 335:IF D THEN R
    ETURN                                    <221>
326 GOSUB 370:GOTO 324                       <073>
330 J=J+J3:IF J>J2 THEN 374                  <064>
331 GOTO 323                                 <149>
```

Listing 4. »Funprint« demonstriert die Anwendung von »Prograf«

```
335 D=1:IF PEEK(631)=95 THEN D=0:!DRAW:!SC
    N,1                                       <031>
336 RETURN                                    <140>
340 REM *** STRECKE ***                       <069>
341 !LINE,I1*M%,I2*M%,J1*M%,J2*M%:GOTO 374    <092>
342 !ZLINE,I1*M%,I2*M%,I3*M%,J1*M%,J2*M%,J
    3*M%:GOTO 374                             <232>
370 X2=X1:Y2=Y1:Z2=Z1:X1=FN A(X)*M%:Y1=FN
    B(X)*M%:Z1=FN C(X)*M%                     <009>
371 !ZLINE,X1,Y1,Z1,X2,Y2,Z2:RETURN           <005>
374 FJ=1                                      <052>
375 RETURN                                    <179>
380 REM GEMERKTE FUNKTIONEN                   <239>
381 FI=0:IF FZ=0 THEN 40                      <183>
382 !SCN,1:FF=0:GOSUB 54                       <045>
383 FI=FI+1:IF FI>FZ THEN FI=FZ:GOTO 386      <116>
385 GOSUB 390:FJ=0:GOSUB 210:IF FJ THEN 38
    3                                         <161>
386 POKE 198,0:!DRAW:GOTO 166                 <012>
388 IF F%>0 AND F%<>3 THEN POKE 1031+FU%(F
    %),50-(F%=2 OR F%=5)                      <205>
389 RETURN                                    <193>
390 F%=F%(FI):IF F%<4 THEN A$=A$(FI):D$=A$
    :GOSUB 181:!DEF FN A                      <135>
391 IF F%THEN IF F%<4 THEN B$=B$(FI):D$=B$
    :GOSUB 181:!DEF FN B                      <119>
392 IF F%=3 OR F%=2 THEN C$=C$(FI):D$=C$:G
    OSUB 181:!DEF FN C                        <189>
393 I1=I1(FI):I2=I2(FI):I3=I3(FI):J1=J1(FI
    ):J2=J2(FI):J3=J3(FI):RETURN              <092>
400 REM --- DRUCKEN -----------------         <157>
401 GOSUB 701:GR%=0                           <195>
402 CY%=17:GOSUB 711:PRINT,"{3SPACE}AUSDRU
    CK                                        <236>
405 CY%=19:GOSUB 711:PRINT" 320*200{SPACE,
    RVSON,SPACE}0{SPACE,RVOFF,SPACE}SCHNEL
    L{3SPACE}MIT RAND{SPACE,RVSON,SPACE}R{
    SPACE,RVOFF}                              <158>
406 CY%=20:GOSUB 711:PRINT" 640*400{SPACE,
    RVSON,SPACE}1{SPACE,RVOFF,SPACE}KLEIN     <072>
407 CY%=21:GOSUB 711:PRINT"{3SPACE}OPTI-{S
    PACE,RVSON,SPACE}2{SPACE,RVOFF,SPACE}M
    ITTEL{4SPACE}MIT TEXT{SPACE,RVSON,SPAC
    E}T{SPACE,RVOFF}                          <1[illegible]2>
408 CY%=22:GOSUB 711:PRINT"{3SPACE}MIERT{S
    PACE,RVSON,SPACE}3{SPACE,RVOFF,SPACE}G
    ROSS                                      <098>
410 POKE F+770+GR%*40,12:POKE F+793,12*RD%
    :POKE F+873,12*TX%                        <227>
420 POKE 198,0:WAIT 198,1:GET D$:D=ASC(D$)    <192>
421 IF D$="R"THEN RD%=1-RD%:GOTO 402          <103>
422 IF D$="T"THEN TX%=1-TX%:GOTO 402          <059>
423 IF D>47 AND D<52 THEN GR%=D-48:GOTO 40
    2                                         <057>
425 GOTO 430                                  <171>
426 PRINT,"{GREY 2}DRUCKER EINSCHALTEN{BLA
    CK,UP}                                    <215>
427 POKE 198,0:WAIT 198,1:GET D$              <103>
428 IF D$<>CHR$(13)AND D$<>"D"THEN 35         <076>
430 OPEN 1,4,1:CLOSE 1:IF ST<0 THEN 426       <005>
440 !SCN,1:!LOC,0,0,0,638,0,398               <176>
446 !LINE,0,0,638,0,RD%*255:!LINE,0,0,0,-3
    98,RD%*255                                <189>
447 !LINE,0,-398,638,-398,RD%*255:!LINE,63
    8,0,638,-398,RD%*255                      <013>
448 !LOC,OX%,OY%,2,636,2,396                  <069>
450 OPEN 2,4,1:REM 1= SEK.ADR. FUER GROSS/
    GRAFIK                                    <183>
460 !COPY,GR%,14:REM 14=SEK.ADR. FUER LINE
    ARKANAL                                   <030>
468 IF TX%=0 OR FZ=0 THEN 492                 <159>
470 FOR I=1 TO FZ:IF F%(I)=0 THEN PRINT#2,
    "F(X) = "+A$(I):GOTO 485                  <042>
471 IF F%(I)<4 THEN PRINT#2,"X = "+A$(I):P
    RINT#2,"Y = "+B$(I)                       <170>
473 IF F%(I)=2 OR F%(I)=3 THEN PRINT#2,"Z
    = "+C$(I)                                 <008>
474 IF F%(I)>3 THEN 480                       <088>
475 PRINT#2,"I VON "+STR$(I1(I))+" BIS "+S
    TR$(I2(I))+", SCHRITTW."+STR$(I3(I))      <099>
476 IF F%(I)<3 THEN 485                       <252>
477 PRINT#2,"J VON "+STR$(J1(I))+" BIS "+S
    TR$(J2(I))+", SCHRITTW."+STR$(J3(I))      <013>
478 GOTO 485                                  <208>
480 PRINT#2,"("+STR$(I1(I))+","+STR$(I2(I)
    );                                        <080>
481 IF F%(I)=5 THEN PRINT#2,","+STR$(I3(I)
    );                                        <030>
482 PRINT#2,"){2SPACE}-{2SPACE}("+STR$(J1(
    I))+","+STR$(J2(I));                      <206>
483 IF F%(I)=5 THEN PRINT#2,","+STR$(J3(I)
    );                                        <048>
484 PRINT#2,")"                               <058>
485 PRINT#2:NEXT                              <205>
486 PRINT#2,"URSPRUNG ("STR$(OX%)","STR$(O
    Y%)"), EINHEIT"STR$(M%);                  <186>
487 IF F%<2 OR F%=4 THEN 492                  <087>
488 PRINT#2,", PROJEKTIONSZENTRUM ("STR$(P
    %(1))","STR$(P%(2))","STR$(P%(3))")"      <064>
490 PRINT#2,"WINKEL: ("STR$(R%(1))","STR$(
    R%(2))","STR$(R%(3))"), ";                <216>
491 PRINT#2,"VERSCHIEBUNG: ("STR$(T%(1))",
    "STR$(T%(2))","STR$(T%(3))") "            <022>
492 PRINT#2:CLOSE 2:!SCN,0:GOTO 35            <062>
500 REM --- STRINGS,DATA,FELDER -----         <092>
510 NN$="NAME":KS$="+-£@*↑:;=K":DEF FN KS(
    X)=F+272+3*X+71*INT(X/3)-2*(X=9)          <153>
521 DEF FN E(X)=X-INT(X/10)*10+48             <034>
522 DEF FN Z(X)=32-(16+INT((X-INT(X/100)*1
    00)/10))*(X>9)                            <026>
523 DEF FN H(X)=32-(16+INT(X/100))*(X>99):
    DEF FN V(X)=32-13*(X<0)                   <134>
531 DATA 241,281,281,321,361,361:FOR I=0 T
    O 5:READ FU%(I):NEXT                      <144>
541 DATA,"SQR(..)","LOG(..)","SQR(..) UND
    LOG(..)","(..)↑"                          <011>
542 DATA"SQR(..) UND (..)↑","LOG(..) UND (
    ..)↑","SQR(..) UND LOG(..) UND (..)↑"     <222>
543 FOR I=0 TO 7:READ DT$(I):NEXT             <208>
551 DIM P1(4),P2(4),P3(4),PF%(5),R%(3),P%(
    3),T%(3)                                  <026>
552 FOR I=0 TO 4:READ P1(I),P2(I),P3(I):NE
    XT:FOR I=0 TO 5:READ PF%(I):NEXT          <255>
554 DATA-240,-1100,200,-180,-1100,300,0,-1
    100,330,180,-1100,300,240,-1100,200       <169>
555 DATA 291,252,214,256,297,259              <104>
560 DIM F%(35),A$(35),B$(35),C$(35),I1(35)
    ,I2(35),I3(35),J1(35),J2(35),J3(35)       <114>
561 RETURN                                    <111>
600 REM --- UNTERPROGRAMME -----------        <159>
610 REM URSPRUNG                              <040>
611 GOSUB 701:PRINT:PRINT" KOORDINATEN DES
     URSPRUNGS:"                              <168>
612 POKE 198,0:CY%=19:GOSUB 711:PRINT" X =
    {SPACE,GREY 2}";OX%;"{LEFT,4SPACE}"       <129>
613 CY%=21:GOSUB 711:PRINT"{BLACK,SPACE}Y
    ={SPACE,GREY 2}";OY%;"{BLACK,LEFT,4SPA
    CE}"                                      <093>
614 IF PEEK(203)<>64 THEN 616                 <180>
615 WAIT 198,1:GET D$                         <042>
616 IF D$="X"THEN OX%=OX%-20:GOTO 612         <079>
617 IF D$="Y"THEN OY%=OY%+20:GOTO 612         <022>
618 IF D$="Y"THEN OY%=OY%-20:GOTO 612         <089>
619 IF D$="X"THEN OX%=OX%+20:GOTO 612         <016>
648 RETURN                                    <198>
649 OX%=240*(KS%-INT(KS%/3)*3)+80:OY%=140*
    INT(KS%/3)+60:RETURN                      <188>
650 REM PROJEKTIONSZENTRUM                    <101>
651 GOSUB 701:PRINT:PRINT" KOORDINATEN DES
     PROJEKTIONSZENTRUMS:"                    <179>
652 POKE 198,0:FOR I=1 TO 3:CY%=17+2*I:GOS
    UB 711:PRINT"{BLACK}"CHR$(87+I)" ={SPA
    CE,GREY 2}"P%(I);                         <160>
653 PRINT"{BLACK,LEFT,4SPACE}":NEXT           <235>
655 IF PEEK(203)<>64 THEN 657                 <224>
656 WAIT 198,1:GET D$:FOR II=1 TO 6           <115>
657 FOR I=1 TO 6                              <122>
658 IF D$=MID$("XXYYZZ",I,1)THEN D=INT((I+
    1)/2):P%(D)=P%(D)+20+40*(I/2=D):GOTO 6
    52                                        <224>
659 NEXT:RETURN                               <090>
699 P%(1)=P1(PZ):P%(2)=P2(PZ):P%(3)=P3(PZ)
    :RETURN                                   <093>
700 REM EINGABEFELD LOESCHEN                  <098>
701 FOR I=16 TO 24:POKE 781,I:SYS 59903:NE
    XT:CX%=0:CY%=16:GOSUB 711:RETURN          <205>
710 REM CURSOR                                <142>
711 POKE 211,CX%:POKE 214,CY%:SYS 58640:RE
    TURN                                      <083>
720 REM INPUT                                 <201>
```

```
721 IF D$<>""THEN IF LEFT$(D$,1)=" "THEN D
    $=RIGHT$(D$,LEN(D$)-1)                      <150>
722 D$=D$+"{67SPACE}"                           <127>
723 X2%=0:D$=LEFT$(D$,IL)                       <228>
725 GOSUB 711:PRINT"{RVSON}"+D$+"{RVOFF,SP
    ACE}";:X1=F+CX%+40*CY%+X2%                  <076>
730 POKE X1,12:GET E$:IF E$=""THEN 730          <197>
731 D=ASC(E$):IF D=13 THEN 760                  <153>
732 IF D=29 THEN IF X2%<IL-1 THEN X2%=X2%+
    1:POKE X1,0:X1=X1+1:GOTO 730                <154>
734 IF D=157 THEN IF X2%THEN X2%=X2%-1:POK
    E X1,0:X1=X1-1:GOTO 730                     <013>
735 IF D<>20 THEN 737                           <226>
736 IF X2%THEN X2%=X2%-1:D$=LEFT$(D$,X2%)+
    MID$(D$,X2%+2,IL-1-X2%)+" ":GOTO 725        <213>
737 IF D=148 THEN D$=LEFT$(D$,X2%)+" "+MID
    $(D$,X2%+1,IL-1-X2%):GOTO 725               <247>
738 IF D<59 THEN IF D>39 THEN IF D<>44 THE
    N 745                                       <195>
740 IF D>63 THEN IF D<89 THEN IF IL=66 OR
    IL=14 THEN 745                              <026>
741 IF D=32 OR D=94 THEN 745                    <227>
742 IF D=222 THEN E$=CHR$(255):GOTO 745         <078>
743 GOTO 725                                    <147>
745 D$=LEFT$(D$,X2%)+E$+MID$(D$,X2%+2,IL-1
    -X2%):IF X2%<IL-1 THEN X2%=X2%+1            <166>
750 GOTO 725                                    <154>
760 POKE X1,0:I=1                               <079>
761 IF MID$(D$,I,1)<>" "THEN IF I<LEN(D$)+
    1 THEN I=I+1:GOTO 761                       <099>
762 D$=LEFT$(D$,I-1):IF D$=""THEN D$="0"        <144>
763 RETURN                                      <057>
770 REM FKT. LOESCHEN                           <102>
771 CX%=20:CY%=14:GOSUB 711:PRINT"{GREY 2}
    B{BLACK}ILD{GREY 2}T{BLACK}ERM{GREY 2}
    G{BLACK}ES":POKE 198,0:WAIT 198,1           <089>
772 GET D$:IF D$="G"THEN:!CLR:FZ=0:FI=0:KR
    %=0:GOSUB 801:GOTO 778                      <021>
773 IF D$="B"THEN:!CLR:KR%=0:GOTO 778           <139>
774 IF D$<>"T"OR FZ=0 THEN 778                  <124>
775 IF FI=FZ THEN FI=FI-1:GOTO 777              <102>
776 II=FI:FI=0:GOSUB 205:FI=FZ:GOSUB 390:F
    I=II:GOSUB 205:FI=0:GOSUB 390:FI=II         <104>
777 FZ=FZ-1                                     <137>
778 GOSUB 711:PRINT"{2SPACE}LOESCHEN ":RET
    URN                                         <242>
780 REM ANZEIGE EINGABEFELD                     <119>
781 GOSUB 701:IF F%>3 THEN 791                  <179>
782 IF F%=0 THEN F1$="F(X) = ":F2$="":F3$=
    "":GOTO 785                                 <130>
783 IF F%=3 THEN F1$="{LEFT}X(I,J)= ":F2$=
    "{LEFT}Y(I,J)= ":F3$="{LEFT}Z(I,J)= ":
    GOTO 785                                    <236>
784 F1$="X(I) = ":F2$="Y(I) = ":F3$="":IF
    F%=2 THEN F3$="Z(I) = "                     <188>
785 IF F%THEN PRINT" I VON ";I1;"{LEFT}";T
    AB(14);"BIS ";I2;"{LEFT}";TAB(25);"STE
    P ";I3                                      <249>
786 CY%=17:GOSUB 711                            <075>
787 IF F%=3 THEN PRINT" J VON ";J1;"{LEFT}
    ";TAB(14);"BIS ";J2;"{LEFT}";TAB(25);"
    STEP ";J3                                   <016>
788 CY%=18:GOSUB 711:PRINT" "F1$;A$:IF F%T
    HEN CY%=20:GOSUB 711:PRINT" "F2$;B$         <066>
789 IF F%>1 THEN CY%=22:GOSUB 711:PRINT" "
    F3$;C$;                                     <221>
790 RETURN                                      <086>
791 PRINT" PUNKT A{8SPACE}PUNKT B               <191>
792 CY%=18:GOSUB 711:PRINT" ";I1;TAB(16);J
    1                                           <181>
793 CY%=20:GOSUB 711:PRINT" ";I2;TAB(16);J
    2                                           <002>
794 IF F%=5 THEN CY%=22:GOSUB 711:PRINT" "
    ;I3;TAB(16);J3                              <123>
795 RETURN                                      <091>
800 REM EINHEIT,WINKEL,NR.                      <252>
801 POKE 1529,FN E(FI):POKE 1528,FN Z(FI):
    II=1481:I=M%:GOTO 805                       <185>
802 FOR D=1 TO 3:II=1286+D*40:I=R%(D):GOSU
    B 805:I=T%(D):GOSUB 806:NEXT:RETURN         <198>
805 POKE II+2,FN E(I):POKE II+1,FN Z(I):PO
    KE II,FN H(I):RETURN                        <241>
806 POKE II+6,FN E(ABS(I)):POKE II+5,FN Z(
    ABS(I)):POKE II+4,FN V(I):RETURN            <109>
900 REM PROGRAMMENDE                            <073>
902 REM (W)RITTEN BY B. HAFNER                  <073>
904 REM UEBERARBEITET VON N. HEUSLER            <084>
905 REM  (16.7.89)                              <216>
906 REM I.A. MARKT & TECHNIK VERLAG             <222>
908 REM (C) REDAKTION SONDERHEFTE 64'ER         <187>
```

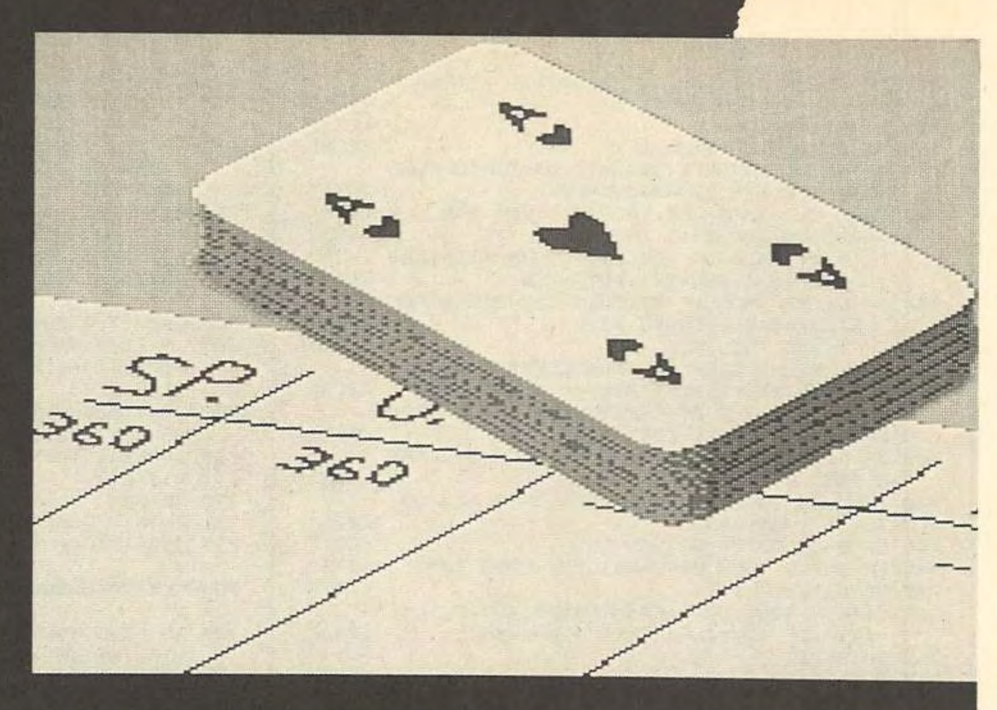
Listing 4. (Schluß)

```
1 REM    DEMOPROGRAMM 'PROGRAF'                 <119>
2 REM -----------------------------------       <153>
3 REM PROGRAMMIERT VON NIKOLAUS HEUSLER         <205>
4 REM (C) SONDERHEFT 64'ER     NH-160789        <117>
5 REM -----------------------------------       <156>
6 REM WE APOLOGIZE FOR ANY INCONVENIENCE        <236>
10 POKE 56,67:CLR:POKE 53280,11:POKE 0,111      <124>
11 IF PEEK(17803)+PEEK(22956)<>33 THEN SYS
   57812"PROGRAF",8,1:POKE 780,0:SYS 6549
   3                                            <017>
12 POKE 777,67                                  <217>
14 GOTO 100                                     <198>
20 POKE 53280,.:POKE 198,.:WAIT 198,1:POKE
   198,.:POKE 53280,11:RETURN                   <005>
100 PRINT"{CLR,SPACE,DOWN}DEMOPROGRAMM 'PR
    OGRAPH'{17SPACE}VON NIKOLAUS HEUSLER        <119>
102 PRINT"{DOWN,SPACE}IMMER, WENN DIE RAHM
    ENFARBE AUF SCHWARZ                         <191>
104 PRINT"{UP,SPACE}SCHALTET, IST DIE GRAF
    IK FERTIG.                                  <191>
106 PRINT"{DOWN,SPACE}DANN WOLLEN SIE BITT
    E EINE TASTE DRUEK- KEN.                    <203>
108 GOSUB 20                                    <054>
110 !SCN,1:!CLR,11,1                            <087>
120 !LOC,320,200:OX=160:OY=0                    <124>
130 FOR I=.TO 63 STEP.1:X=40*COS(I)*(4+SIN
    (I*3.3)):Y=40*SIN(I)*(3+SIN(I*3.3))         <156>
140 !LINE,X,Y,OX,OY:OX=X:OY=Y:NEXT:GOSUB 2
    0                                           <163>
150 !CLR:!TRANS,0,0,0:!ROT,0,0,0                <080>
160 !LOC,320,340:E=40:!PROZ,180,-600,300        <130>
170 FOR I=-26 TO 14 STEP.5:FL=.:FOR J=-4 T
    O 30 STEP.6                                 <111>
180 X=I:Y=J:Z=SIN(I-J+SIN(I+J))/(3+J/3)         <225>
190 IF FL THEN:!ZLINE,X*E,Y*E,Z*E,OX*E,OY*
    E,OZ*E                                      <226>
195 FL=1:OZ=Z:OY=Y:OX=X                         <135>
200 NEXT J,I                                    <015>
210 FL=.:FOR I=-7 TO 7 STEP.2                   <131>
220 X=I:A=LOG(3+I*I)+3:B=LOG(1+I*I)+3           <235>
230 IF FL THEN:!ZLINE,X*E,0,A*E,OX*E,0,OA*
    E                                           <188>
232 IF FL THEN:!ZLINE,X*E,0,B*E,OX*E,0,OB*
    E                                           <207>
234 FL=1:OX=X:OA=A:OB=B:NEXT:GOSUB 20           <145>
240 !CLR:!PROZ,180,-1100,300:E=55:!LOC,320
    ,60:!TRANS,-3*E,0,-2*E                      <186>
250 FOR I=-3 TO 3 STEP.5:FL=.:FOR J=-3 TO
    3 STEP.5                                    <101>
260 X=I:Y=J:Z=-2/SQR(I*I+J*J+.1)                <022>
270 IF FL THEN:!ZLINE,X*E,Y*E,Z*E,OX*E,OY*
    E,OZ*E                                      <052>
275 FL=1:OZ=Z:OY=Y:OX=X:NEXT:NEXT               <095>
280 FOR I=.5 TO 3 STEP.3:FL=0:FOR J=0 TO 6
    .29 STEP.314                                <137>
290 X=I*COS(J):Y=I*SIN(J):Z=-2/I                <032>
300 IF FL THEN:!ZLINE,X*E,Y*E,Z*E,OX*E,OY*
    E,OZ*E                                      <082>
305 FL=1:OZ=Z:OY=Y:OX=X:NEXT:NEXT               <125>
310 GOSUB 20:END                                <105>
```

Listing 5. Ein einfaches Demoprogramm für »Prograf« (bitte bei nicht aktiviertem »Prograf« mit dem Checksummer eingeben)

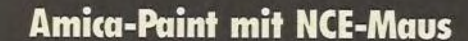
NEUES FÜR AMICA PAINT

Amica-Paint, unser tolles Malprogramm aus dem Sonderheft 27, läßt sich noch weiter verbessern. Hier finden Sie einige brandneue Erweiterungen.



Einige der hier vorgestellten Erweiterungen können unter Amica-Paint als Erweiterung nachgeladen werden. Diese erkennen Sie an den Zeichen »[W]« im Filenamen. Dazu wählen Sie im Hauptmenü die Funktion »Sonstiges« (Taste <O>). Betätigen Sie anschließend die Taste <E>, listet Amica-Paint alle Erweiterungen der eingelegten Diskette auf. Über den dabei angezeigten Buchstaben wird die gewünschte Erweiterung gestartet. Es ist daher notwendig, daß Sie die Listings, die Sie aus diesem Artikel abtippen, auf Ihrer Amica-Paint-Systemdiskette speichern, ggf. auf der Rückseite. Der Betrieb der Programme aus diesem Artikel ohne Amica-Paint ist leider nicht möglich.

Neben einigen Erweiterungen stellen wir Ihnen zwei neue Eingabetreiber vor, die vor dem Start von Amica-Paint installiert werden können. Es handelt sich um einen Treiber für die NCE-Maus und einen stark verbesserten Treiber für die Maus 1351 von Commodore, die jetzt in drei verschiedenen Geschwindigkeiten betrieben werden kann. Auch zwei neue Zeichensätze können wir Ihnen offerieren. Da müßte doch eigentlich für jeden Amica-Paint-Liebhaber etwas dabeisein ...



Amica-Paint mit NCE-Maus

Beginnen wir mit den beiden Maustreibern. Grundsätzlich gehen Sie bitte wie folgt vor: Tippen Sie das entsprechende File (Listing 10 und/oder 11) ab und speichern Sie es auf der Amica-Paint-Diskette. Dieses File stellt später die Verbindung von Amica-Paint und der Maus her (sog. »Treiber«). Dann muß noch das Programm »Eingabegeräte« aus dem Sonderheft 27 auf die neuen Treiber eingestellt werden. Dazu ergänzen Sie einfach die DATA-Zeilen entsprechend. Sie können aber auch das Installationsprogramm völlig neu eingeben (sicherer), dazu tippen Sie Listing 15 ein und speichern es.

Listing 10 (bitte mit dem MSE eingeben, siehe Seite 159) ermöglicht das Betreiben von Amica-Paint mit der NCE-Maus. Nach dem Starten des Programmes »EINGABEGERAETE« (entweder wie beschrieben geändert oder abgetippt mit dem Checksummer als Listing 15) kann nun neben Joystick und der Commodore-Maus 1351 auch die NCE-Maus angewählt werden. Da diese normalerweise die Tastatur blockiert, muß ein Adapter (bei der Firma Scanntronic für ca. 25 Mark erhältlich) zwischen die Maus und den C 64 geschaltet werden. Die NCE-Maus wird an Port 2 angeschlossen. Der rechte Mausknopf entspricht dem Feuerknopf des Joysticks und dient zum Ausführen von Funktionen. Der linke Knopf hingegen führt zum Abbruch eines Befehls. Er entspricht der STOP-Taste.

Dreimal 1351

Aber auch für die Maus 1351 gibt es einen neuen Treiber. Er löst das Problem des zu schnellen Mauszeigers: Die Geschwindigkeit wird wahlweise auf die Hälfte oder ein Viertel reduziert. Dazu müssen die Listings 11 bis 14 mit dem MSE eingegeben und auf der Amica-Paint Diskette gespeichert werden. Nun das Programm »EINGABEGERAETE« laden und folgende Zeile eingeben:

```
410 DATA "MAUS 1351 LMS","MAUS LMS"
```

Dieses Programm wird nun nach dem Löschen der alten Version auf der Diskette gespeichert. Sie können aber auch hier Listing 15 eingeben und ohne Änderung als neue Version speichern. Nach dem Start des Installationsprogrammes und Anwählen des Menüpunkts »MAUS 1351 LMS« (»LMS« steht für »Langsam, Mittel, Schnell«) wird der neue Maustreiber installiert.

Nach dem Starten von Amica-Paint merken Sie noch keinen Unterschied. Sie können jedoch jetzt im Erweite-

rungsmenü (Taste <O> und <E>) drei verschiedene Geschwindigkeiten wählen:
SCHNELL: ursprüngliche Geschwindigkeit
NORMAL: halbe Geschwindigkeit
LANGSAM: ein Viertel der alten Geschwindigkeit.

Hiermit ist ein sehr präzises Arbeiten mit der Maus möglich. Bitte beachten Sie, daß beim Anwählen von einem der drei Erweiterungen unbedingt der neue Maustreiber installiert sein muß und nach dem Aufruf der gewünschte Zeichenbefehl noch anzuwählen ist.

Drehen mit Amica-Paint

Da die Drehfunktion von Amica-Paint für beliebige Winkel ausgelegt ist und infolgedessen immer mit einigen Ungenauigkeiten arbeitet, wurde speziell für das Drehen um den Winkel von 90 Grad eine Erweiterung entwickelt, die genauer ist. Das File »[W]DREHEN 90 GR.« (Listing 4) geben Sie bitte mit dem MSE ein und speichern es auf die Amica-Paint-Diskette. Die neue Drehfunktion kann jetzt unter dem Malprogramm im Erweiterungsmenü aufgerufen werden. Im Grafikeditor wird einfach der zu drehende Block festgelegt. Danach wählen Sie noch die Position des um 90 Grad im Uhrzeigersinn gedrehten Blocks.

Der Amica-Paint-Schnellader



Auch diese Erweiterung (Listing 3) soll mit dem MSE eingegeben werden. Sie entspricht dem schon auf der Systemdiskette enthaltenen MOST ACCESS.

Durch das Anwählen der Erweiterung »[W]Schnellader« installiert sich ein Software-Speeder, mit dem Bilder und Erweiterungen ungefähr fünfmal schneller geladen werden als gewöhnlich. Dies ist für all jene interessant, die keinen eingebauten Diskettenbeschleuniger besitzen.

Wenn Bilder auf Wanderschaft gehen

Farbbilder liegen manchmal als 33 oder 32 Blocks lange Files vor. Es handelt sich dabei um Bitmaps ohne Farbinformationen. Im ganzen Bild werden nur vier verschiedene Farben verwendet. Zur Umwandlung solcher Bilder in das Amica-Paint-Format dient das Programm »ALLWAND/F« (Listing 7, bitte mit dem Checksummer eingeben). Das File »ALLWAND.C000« (Listing 8, bitte mit dem MSE eingeben) wird nachgeladen. Es verwandelt alle auf einer Diskette befindlichen Bilder in das Amica-Paint-Format. Auf der Diskette dürfen sich daher nur Bilder, keine anderen Dateien befinden. Manchmal liegt auch eine Diskette mit mehreren Koala-Bildern vor, die alle in das Amica-Paint-Format umgewandelt werden sollen. Diese Aufgabe erledigt das Programm »ALLWAND« (Listing 9).

Amica-Paint-Bilder zeigen

Das alte Programm »SHOW PIC.52480« aus dem Sonderheft 27, das zum Anzeigen von Bildern dient, hatte den Schönheitsfehler, daß beim Umschalten zwischen zwei Bildern für kurze Zeit ein wüster »Pixel-Salat« auf dem Bildschirm erschien. Dieses Programm wurde überarbeitet. Die Bilder scrollen nun sauber über den Bildschirm. Um eine ansprechende »Diashow« zu realisieren, benötigt man nun nicht mehr unbedingt die eingebaute Diashowfunktion von Amica-Paint. Es genügt, wenn man die Files »SHOW« (Listing 1, bitte mit dem Checksummer eingeben) und »SHOW PIC.52480« (Listing 2) auf die Bilderdiskette kopiert. Nach dem Starten von »SHOW« werden nun alle Bilder auf der Diskette nacheinander angezeigt.

Neue Charakter

An dieser Stelle können wir Ihnen auch noch zwei neue Zeichensätze anbieten, die nur jeweils vier Blocks lang sind. Geben Sie diese (Listings 5 und 6) mit dem MSE ein und speichern sie auf der Systemdiskette. Sie können dann wie gehabt nachgeladen werden. Auf der Service-Diskette zum Heft finden Sie außerdem noch zwei weitere Zeichensätze (»Mini« und «Profi«), die bereits im Stammheft 8/88 veröffentlicht wurden. Ebenfalls enthalten sind die beiden Erweiterungen »[W]Texteditor« und »[W]Text« derselben Ausgabe. »[W] Text« sollte dabei auf jeden Fall auf der Diskette mit allen anderen Erweiterungen gespeichert sein.

Das waren nun also unsere Erweiterungen zu Amica-Paint. Wir hoffen, Sie haben nun noch mehr Vergnügen mit diesem netten Malprogramm. Wenn Sie noch mehr solche Erweiterungen entwickelt haben – immer her damit!

(Oliver Stiller/Horst Makitta/Nikolaus Heusler/ag)

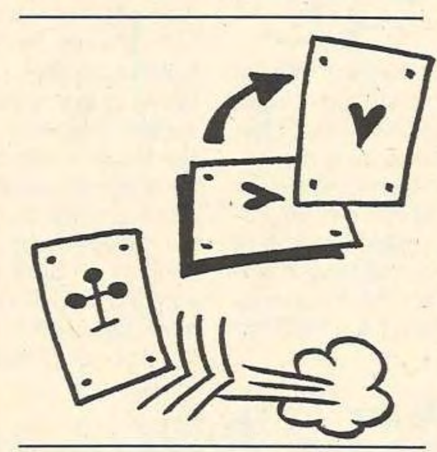
Info zum Adapter für die NCE-Maus: Scanntronic, Parkstraße 38, 8011 Zorneding, Telefon 08106/22570

```
100 REM OLIVER STILLER                  <154>
110 REM                                 <172>
120 IF L=0 THEN L=1 :LOAD "SHOW PIC.52480"
    ,8,1                                <094>
130 DIM NA$(144):C3$=CHR$(34)           <107>
140 PRINT"{2DOWN}AMICA PAINT DIASHOW"   <068>
150 PRINT"DIESES PROGRAMM ZEIGT ALLE AMICA
     PAINT{2SPACE}BILDER EINER DISK AN." <217>
160 PRINT"{2DOWN}BITTE DISK EINLEGEN UND T
    ASTE DRUECKEN":WAIT 198,1:GET A$    <090>
170 FL=0:IF A$=" " THEN FL=1            <238>
180 GOSUB 340                           <204>
190 :                                   <166>
200 IF PO=0 THEN GOTO 160               <229>
210 :                                   <186>
220 FOR T=0 TO PO-1                     <112>
230 IF FL=0 THEN PRINT"{DOWN}LADE BILD:";C
    3$;NA$(T);C3$                       <000>
240 SYS 52480,8,NA$(T),1                <177>
250 FOR G=1 TO 1400:GET A$:IF A$="" THEN N
    EXT                                 <155>
260 IF FL THEN NEXT T                   <232>
270 SYS 52480,8,"",0                    <018>
280 IF FL=0 THEN NEXT T                 <157>
290 GOTO 160                            <060>
310 REM                                 <118>
320 REM DIRECTORY ZEIGEN                <154>
330 REM                                 <138>
340 CLOSE 1:OPEN 1,8,0,"$:[B]*":GET#1,A$,A
    $:PO=0                              <056>
350 PRINT"{DOWN}BILDER DER DISKETTE:"   <096>
360 FOR T=0 TO 7:GET#1,A$,A$,A$,A$:NEXT <077>
370 :                                   <092>
380 GET#1,A$,A$,A$,A$:IF ST THEN 410    <103>
390 GET#1,X$:IF X$=C3$ THEN AN$="":GOTO 42
    0                                   <252>
400 IF ST=0 THEN 390                    <057>
410 CLOSE 1:RETURN                      <027>
420 GET#1,Z$:IF Z$<>C3$ THEN AN$=AN$+Z$:GO
    TO 420                              <096>
430 PRINT C3$;AN$;C3$:NA$(PO)=AN$:PO=PO+1 <126>
440 GET#1,X$:IF X$<>"" AND ST=0 THEN 440  <127>
450 GOTO 380                            <012>
```

Listing 1. Das Programm »SHOW« zeigt eine Diashow. Bitte geben Sie es mit dem Checksummer (Seite 159) ein.

```
Name : show pic.52480     cd00 cf6f
-----------------------------------
cd00 : a9 00 8d 6d cf 8d 20 d0  46
cd08 : 20 48 ce 20 fd ae 20 9e  17
cd10 : b7 a9 00 a0 00 20 ba ff  9c
cd18 : 20 fd ae 20 9e ad 20 a3  06
cd20 : b6 a6 22 a4 23 48 20 bd  b7
cd28 : ff 20 fd ae 20 9e b7 8e  7f
cd30 : 6e cf a2 00 a0 a0 86 fd  54
cd38 : 84 fe 68 f0 1d a9 00 20  d3
cd40 : d5 ff 90 03 4c f9 e0 ad  0d
cd48 : 6e cf 29 01 0d 72 ce f0  8a
cd50 : 06 20 73 ce ee 6d cf 20  f7
cd58 : 00 cf ad 6d cf d0 13 ae  86
cd60 : 6e cf e0 02 f0 08 ad 72  19
cd68 : ce cd 6e cf d0 01 60 20  89
cd70 : 73 ce ad 6e cf d0 06 20  5f
cd78 : 83 cd 4c 7e ce 20 c3 cd  5d
cd80 : 4c 7e ce a0 00 8c 11 d0  1d
cd88 : b9 00 c8 99 00 d8 b9 00  54
cd90 : c9 99 00 d9 b9 00 ca 99  5b
cd98 : 00 da b9 00 cb 99 00 db  b5
cda0 : c8 d0 e5 a9 15 8d 18 d0  3f
cda8 : ad 00 cc 8d 21 d0 a9 c8  0b
cdb0 : 8d 16 d0 a9 1b 8d 11 d0  b6
cdb8 : a9 97 8d 00 dd a9 00 8d  d6
cdc0 : 72 ce 60 a0 00 b9 00 c0  15
cdc8 : 99 00 d8 b9 00 c1 99 00  43
cdd0 : d9 b9 00 c2 99 00 da b9  57
cdd8 : 00 c3 99 00 db 88 d0 e5  31
cde0 : a9 1f 8d 18 d0 ad 01 cc  97
cde8 : 8d 21 d0 a9 d8 8d 16 d0  63
cdf0 : a9 3b 8d 11 d0 a9 94 8d  84
cdf8 : 00 dd a9 01 8d 72 ce 60  da
ce00 : a2 00 81 fb e6 fb d0 02  18
ce08 : e6 fc 60 a2 00 a1 fd e6  ac
ce10 : fd d0 02 e6 fe 60 20 0b  5c
ce18 : ce c9 c2 f0 06 20 00 ce  98
ce20 : 4c 16 ce 20 0b ce c9 00  7d
ce28 : f0 1d a8 20 0b ce 48 98  4e
ce30 : aa 68 a0 00 91 fb c8 ca  e8
ce38 : d0 fa 98 18 65 fb 85 fb  f3
ce40 : 90 d4 e6 fc 4c 16 ce 60  05
ce48 : ad 72 ce f0 01 60 a0 00  96
ce50 : b9 00 d8 99 00 c8 b9 00  a0
ce58 : d9 99 00 c9 b9 00 da 99  71
ce60 : 00 ca b9 00 db 99 00 cb  56
ce68 : c8 d0 e5 ad 21 d0 8d 00  96
ce70 : cc 60 00 a9 00 20 80 ce  42
ce78 : a9 0b 8d 11 d0 60 a9 01  e5
ce80 : 8d ff ce 78 ae ff ce bd  71
ce88 : fd ce 8d f8 ce ad ff 3f  48
ce90 : 48 a9 ff 8d ff 3f 8d ff  8e
ce98 : ff ad 12 d0 c9 2e d0 f9  52
cea0 : ad 11 d0 30 f4 ad 11 d0  b3
cea8 : 29 e0 09 12 8d 11 d0 ac  c4
ceb0 : f8 ce ae 12 d0 ad 11 d0  5e
ceb8 : 48 29 f0 85 fb 68 29 1f  67
cec0 : 18 69 01 c9 18 d0 02 a9  6a
cec8 : 00 05 fb 8d 11 d0 ec 12  6b
ced0 : d0 f0 fb 88 d0 dc ad f8  c5
ced8 : ce 18 ae ff ce 7d f9 ce  bc
cee0 : 8d f8 ce dd fb ce d0 b1  35
cee8 : 68 8d ff 3f ad 11 d0 29  f8
cef0 : e0 09 1b 8d 11 d0 58 60  87
cef8 : 00 02 fe d3 00 01 d2 01  89
cf00 : 78 a2 00 86 01 a0 d0 86  00
cf08 : fb 84 fc 20 16 ce a0 00  e3
cf10 : b9 40 ef 99 00 c4 b9 40  a6
cf18 : f0 99 00 c5 b9 40 f1 99  26
cf20 : 00 c6 b9 40 f2 99 00 c7  85
cf28 : b9 28 f3 99 00 c0 b9 28  63
cf30 : f4 99 00 c1 b9 28 f5 99  11
cf38 : 00 c2 b9 28 f6 99 00 c3  d0
cf40 : c8 d0 cd ad 10 f7 8d 01  92
cf48 : cc a2 20 a9 ef a0 00 85  b2
cf50 : fe 84 fd a9 ff 85 fc 84  6e
cf58 : fb b1 fd 91 fb c8 d0 f9  1b
cf60 : c6 fe c6 fc ca d0 f2 a9  49
cf68 : 37 85 01 58 60 00 00 00  b3
```

Listing 2. Dieses Programm stellt die verbesserte Bild-Einblend-Routine dar. Bitte geben Sie sie mit dem MSE (Seite 159) ein.



```
Name : [w]schnellader     4000 41e0
-----------------------------------
4000 : a0 37 b9 40 40 99 34 03  5a
4008 : 88 10 f7 a0 4e b9 80 40  df
4010 : 99 b0 02 88 10 f7 a0 6f  b5
4018 : b9 d0 40 99 00 0b 88 10  17
4020 : f7 a0 97 b9 3f 41 99 3f  67
4028 : ef 88 d0 f7 a0 0b a2 00  7b
4030 : 8c 31 03 8e 30 03 00 48  93
4038 : 40 ce 27 2e 42 43 53 20  3b
4040 : 4c 3a 03 4c 56 03 48 58  43
4048 : ad 11 d0 09 10 8d 11 d0  26
4050 : a9 2a 20 c3 ff 68 85 90  09
4058 : 4a 68 68 a9 1d a6 ae a4  31
4060 : af 60 48 a9 2a 20 c3 ff  39
4068 : 68 a8 a9 2a a2 08 20 ba  34
4070 : ff 4c c0 ff 00 55 4d 4c  3e
4078 : 49 4e 20 2e 53 54 59 20  34
4080 : a9 0f 20 37 03 a0 00 84  de
4088 : b0 a2 2a 20 c9 ff a4 b0  a8
4090 : a9 4d 20 d2 ff a9 2d 20  84
4098 : d2 ff a9 57 20 d2 ff 98  89
40a0 : 20 d2 ff a9 06 20 d2 ff  0b
40a8 : a9 20 aa 20 d2 ff 78 c6  ad
40b0 : 01 b9 40 ef e6 01 20 d2  38
40b8 : ff c8 ca d0 f1 84 b0 20  2e
40c0 : cc ff c0 a0 d0 c3 60 de  3a
40c8 : 41 20 eb 53 00 da 40 32  bb
40d0 : a5 b9 85 30 86 ae 84 af  08
40d8 : 20 b0 02 ad 11 d0 29 ef  a3
40e0 : 8d 11 d0 a9 00 20 37 03  43
40e8 : a9 05 a2 6b a0 0b 20 bd  88
40f0 : ff a9 0f 20 37 03 78 a9  4c
40f8 : c3 8d 00 dd 20 4a 0b a8  0f
4100 : 20 4a 0b a6 30 f0 04 84  81
4108 : ae 85 af 20 4a 0b a2 00  f0
4110 : 81 ae e6 ae d0 f5 e6 af  2f
4118 : d0 f1 2c 00 dd 30 fb 08  4b
4120 : a9 c0 4d 00 dd 4a 4a 4d  71
4128 : 00 dd 4a 4a 4d 00 dd 4a  d4
4130 : 4a 4d 00 dd 28 70 03 4c  87
4138 : 34 03 60 4d 2d 45 00 06  b9
4140 : 78 a6 18 a4 19 e0 12 f0  69
4148 : 45 86 06 84 07 ad 00 1c  f9
4150 : 49 08 8d 00 1c a9 80 85  1d
4158 : 00 a6 00 58 78 30 fa e0  6d
4160 : 02 b0 2b a0 00 ad 00 03  0d
4168 : d0 04 ac 01 03 c8 84 8b  25
4170 : a0 02 b9 00 03 a2 02 84  d6
4178 : 8d 20 52 06 a4 8d c8 c4  ce
4180 : 8b d0 ef ac 01 03 ae 00  e8
4188 : 03 d0 be a9 00 2c a9 01  e2
4190 : a2 0a a8 8e 64 06 4a 4a  67
4198 : 4a 4a aa bd 87 06 48 98  65
41a0 : 29 0f aa a9 02 8d 00 18  ed
41a8 : 24 ff bd 87 06 8d 00 18  29
41b0 : 0a 29 0f 8d 00 18 68 8d  42
41b8 : 00 18 0a 29 0f 8d 00 18  fa
41c0 : ea a9 00 8d 00 18 60 0f  91
41c8 : 07 0d 05 0b 03 09 01 0e  91
41d0 : 06 0c 04 0a 02 08 00 20  bf
41d8 : 20 20 20 20 20 20 20 20  d8
```

Listing 3. Der Floppyspeeder zu Amica-Paint ist mit dem MSE (Seite 159) einzugeben und kann als Erweiterung nachgeladen werden

```
Name : [w]drehen 90 gr.   4000 4090
-----------------------------------
4000 : 20 12 70 e6 d2 20 06 70  49
4008 : 20 30 a9 20 8d c0 20 0c  26
4010 : 70 20 24 80 a5 28 a6 25  2a
4018 : e0 64 b0 e9 4a 90 10 0a  11
4020 : 18 69 01 4a a4 34 f0 05  30
4028 : c6 34 4c 2f 40 e6 34 85  18
4030 : 25 8a 0a 85 28 a2 7d a0  9c
4038 : 40 20 27 c0 20 09 70 a5  c2
4040 : 31 85 2f a5 27 85 28 a6  41
4048 : 2f a4 34 20 0f c0 a6 24  b4
4050 : a4 28 48 20 21 90 68 a6  a4
4058 : 24 e6 28 a4 28 20 21 90  b7
4060 : e6 28 e6 2f a5 2f c5 32  49
4068 : 90 dd f0 db c6 34 c6 34  30
4070 : e6 24 a5 24 c5 23 90 c7  9d
4078 : f0 c5 4c 05 40 01 31 03  d5
4080 : 4e 45 55 45 20 50 4f 53  d7
4088 : 49 54 49 4f 4e 20 3f 00  1a
```

Listing 4. Die Funktion zum Drehen von Ausschnitten um 90 Grad stellt ebenfalls eine Erweiterung dar (bitte mit dem MSE eingeben)

```
Name : [z]standard        4480 47c5
-----------------------------------
4480 : 0b 05 04 04 04 04 04 03  06
4488 : 04 04 01 03 04 01 05 03  92
4490 : 04 04 04 03 04 03 04 05  6a
4498 : 05 05 05 04 02 04 02 04  32
44a0 : 04 04 03 03 05 05 04 05  5a
44a8 : 01 03 03 06 05 02 04 01  1f
44b0 : 04 04 03 04 04 04 04 04  70
44b8 : 04 04 04 01 02 04 04 04  38
44c0 : 04 04 05 04 04 04 04 04  01
```

Listing 5. Ein neuer Zeichensatz, der mit dem MSE (Seite 159) eingegeben wird

```
44c8 : 04 04 01 04 04 04 05 04  0c
44d0 : 04 04 05 04 04 05 04 05  1b
44d8 : 05 04 05 04 04 01 04 05  04
44e0 : 05 a1 a8 ae b4 ba c0 c6  67
44e8 : cb d1 d7 d9 de e4 e6 ed  59
44f0 : f2 f8 fe 04 09 0f 14 1a  2c
44f8 : 21 28 2f 36 3c 3f 45 48  23
4500 : 4e 54 5a 5f 64 6b 72 78  57
4508 : 7f 81 86 8b 94 9b 9e a4  45
4510 : a6 ac b2 b7 bd c3 c9 cf  71
4518 : d5 db e1 e7 e9 ec f2 f8  14
4520 : fe 04 0a 11 17 1d 23 29  fe
4528 : 2f 35 3b 3d 43 49 4f 56  d1
4530 : 5c 62 68 6f 75 7b 82 88  14
4538 : 8f 96 9c a3 a9 af b1 b7  fc
4540 : be 45 45 45 45 45 45 45  b9
4548 : 45 45 45 45 45 45 45 45  48
4550 : 45 45 45 46 46 46 46 46  8e
4558 : 46 46 46 46 46 46 46 46  58
4560 : 46 46 46 46 46 46 46 46  60
4568 : 46 46 46 46 46 46 46 46  68
4570 : 46 46 46 46 46 46 46 46  70
4578 : 46 46 46 46 46 46 46 46  78
4580 : 46 47 47 47 47 47 47 47  7f
4588 : 47 47 47 47 47 47 47 47  88
4590 : 47 47 47 47 47 47 47 47  90
4598 : 47 47 47 47 47 47 47 47  98
45a0 : 47 86 39 ef fb 98 80 01  ff
45a8 : 00 0e 17 9f 70 08 88 8e  f0
45b0 : f9 9f e0 08 00 06 f8 8f  e5
45b8 : 60 08 11 17 f9 9f 70 08  b2
45c0 : 00 06 f9 f8 70 08 2d 7e  5a
45c8 : 92 40 60 00 07 f9 9f 71  34
45d0 : e8 88 8e f9 99 90 08 df  dd
45d8 : 9f 24 12 49 78 7c 88 8a  da
45e0 : ec eb 90 08 ff 9f 00 01  e6
45e8 : af d6 b5 a8 01 00 6f 6d  2d
45f0 : a0 50 00 06 f9 9f 60 08  a7
45f8 : 00 0e f9 9f e8 8c 00 07  73
4600 : f9 9f 71 18 00 5f a4 80  b7
4608 : 40 00 06 f4 2f 60 08 5f  3d
4610 : a4 93 20 50 00 09 99 9f  7e
4618 : 60 08 00 01 1d a9 c4 20  0f
4620 : 01 00 01 18 d7 ea 50 01  7d
4628 : 00 01 1d b9 db 88 01 00  2d
4630 : 01 18 ed 4e 23 31 00 0f  5c
4638 : f2 4f f0 08 fa aa f3 ef  c4
4640 : 99 f9 9f e8 8c f5 55 f2  8e
4648 : 96 f9 99 9f 60 08 90 99  f1
4650 : 99 9f 60 08 00 00 00 00  d2
4658 : 00 08 ff a4 80 40 60 b6  ea
4660 : 80 00 00 40 52 bf f5 7f  e2
4668 : ea 50 01 77 eb c7 1e bf  9e
4670 : 70 01 99 32 64 c9 90 08  f4
4678 : 67 a5 26 3e 5e 68 01 e0  f2
4680 : 10 6f 49 33 60 70 d9 92  17
4688 : 5e c0 60 01 16 ce 7c e6  16
4690 : d1 00 00 20 00 08 4f 90  04
4698 : 80 00 01 00 05 e2 00 00  c0
46a0 : f0 00 00 08 01 90 11 32  cf
46a8 : 64 c8 80 08 6f 99 99 9f  fb
46b0 : 60 08 2f 92 49 20 50 6f  e8
46b8 : 11 32 6c f0 08 6f 11 61  1e
46c0 : 1f 60 08 22 26 6a ff 20  4c
46c8 : 08 ff 88 e3 1f 60 08 6f  62
46d0 : 98 ef 9f 60 08 ff 13 22  65
46d8 : 64 40 08 6f 9f 6f 9f 60  01
46e0 : 08 6f 99 f1 13 60 08 33  ff
46e8 : 19 01 45 e2 00 13 6c 63  41
46f0 : 10 08 00 0f 00 f0 00 08  7e
46f8 : 00 8c 63 6c 80 08 6f 91  ce
4700 : 32 20 20 08 09 0e 17 9f  e8
4708 : 70 08 21 1c a5 6f ff 88  2f
4710 : 01 ef 9b eb 9f e0 08 6f  6d
4718 : 98 88 9f 60 08 ef 99 99  82
4720 : 9f e0 08 ff 88 e8 8f f0  21
4728 : 08 ff 88 c8 88 80 08 6f  f6
4730 : 88 b9 9f 60 08 99 9f f9  48
4738 : 99 90 08 ff 9f ff 11 11  7c
4740 : 9f 60 08 9b ae cc eb 90  a7
4748 : 08 88 88 88 8f f0 08 8c  81
4750 : 77 fa c6 31 88 01 99 dd  cf
4758 : fb b9 90 08 6f 99 99 9f  be
4760 : 60 08 ef 99 fe 88 80 08  3a
4768 : 67 a5 29 4a 5e 78 c1 ef  c6
4770 : 99 fc eb 90 08 6f 9c 63  c9
4778 : 9f 60 08 ff c8 42 10 84  31
4780 : 20 01 99 99 99 9f 60 08  e2
4788 : 8c 63 b5 29 c4 20 01 8c  c3
4790 : 63 1a d7 ea 50 01 99 f6  b5
4798 : 66 f9 90 08 8c 76 e2 10  48
47a0 : 84 20 01 ff 32 64 cf f0  dc
47a8 : 08 09 09 99 9f 60 08 ff  c7
47b0 : 9f 09 06 f9 9f 60 08 00  b2
47b8 : 01 ff a9 4a 50 01 a9 1c  58
47c0 : a5 6f ff 88 01 ff 00 00  3e
```

Listing 5. (Schluß)

```
Name : [z]commodore       4480 47ca
-----------------------------------
4480 : 08 06 06 06 05 06 06 05  70
4488 : 06 06 04 05 06 04 07 06  dc
4490 : 06 06 06 06 06 06 06 06  90
4498 : 07 06 06 06 04 07 04 06  79
44a0 : 06 06 02 06 08 06 06 07  c2
44a8 : 04 04 04 08 06 03 06 02  45
44b0 : 07 06 06 06 06 07 06 06  b9
44b8 : 06 06 06 02 03 06 06 06  08
44c0 : 06 06 06 06 06 06 06 06  c0
44c8 : 06 06 04 06 06 06 07 06  4c
44d0 : 06 06 06 06 06 06 06 06  d0
44d8 : 07 06 06 06 06 06 06 08  dd
44e0 : 06 a1 a7 ad b3 b8 be c4  dc
44e8 : c9 cf d5 d9 de e4 e8 ef  e2
44f0 : f5 fb 01 07 0d 13 19 1f  10
44f8 : 25 2c 32 38 3e 42 49 4d  7d
4500 : 53 59 5f 61 67 6f 75 7b  c3
4508 : 82 86 8a 8e 96 9c 9f a5  5a
4510 : a7 ae b4 ba c0 c6 cd d3  b4
4518 : d9 df e5 eb ed f0 f6 fc  14
4520 : 02 08 0e 14 1a 20 26 2c  c0
4528 : 32 38 3e 42 48 4e 54 5b  4d
4530 : 61 67 6d 73 79 7f 85 8b  cf
4538 : 91 98 9e a4 aa b0 b6 bc  d6
4540 : c4 45 45 45 45 45 45 45  bf
4548 : 45 45 45 45 45 45 45 45  48
4550 : 45 45 46 46 46 46 46 46  ce
4558 : 46 46 46 46 46 46 46 46  58
4560 : 46 46 46 46 46 46 46 46  60
4568 : 46 46 46 46 46 46 46 46  68
4570 : 46 46 46 46 46 46 46 46  70
4578 : 46 46 46 46 46 46 46 46  78
4580 : 47 47 47 47 47 47 47 47  80
4588 : 47 47 47 47 47 47 47 47  88
4590 : 47 47 47 47 47 47 47 47  90
4598 : 47 47 47 47 47 47 47 47  98
45a0 : 47 7b 3d f7 c3 17 80 00  ea
45a8 : 07 83 7f 37 c0 03 0c 3e  08
45b0 : cf 3f 80 00 1f 8c 61 e0  dd
45b8 : 00 30 df cf 37 c0 00 07  4a
45c0 : b3 ff 07 80 01 d9 f6 31  62
45c8 : 80 00 07 f3 cd f0 fe 03  ef
45d0 : 0c 3e cf 3c c0 06 0e 66  b8
45d8 : f0 00 c0 31 8c 7e 03 0c  ff
45e0 : 36 f3 6c c0 0e 66 66 f0  d3
45e8 : 00 03 37 ff fa f1 80 00  79
45f0 : 0f b3 cf 3c c0 00 07 b3  e4
45f8 : cf 37 80 00 0f b3 cf ec  2b
4600 : 30 00 07 f3 cd f0 c3 00  e4
4608 : 0f b3 c3 0c 00 00 07 f0  61
4610 : 78 3f 80 00 cf cc 30 c1  f0
4618 : c0 00 0c f3 cf 37 c0 00  14
4620 : 0c f3 cd e3 00 00 03 1e  5e
4628 : bf ef 9b 00 00 0c de 31  04
4630 : ec c0 00 0c f3 cd f1 bc  ed
4638 : 00 0f c6 31 8f c0 fc cc  24
4640 : cc f0 f1 b3 67 cc d9 be  39
4648 : 60 f3 33 33 f0 cd ec f3  ee
4650 : cf 37 80 cc 0c f3 cf 37  83
4658 : 80 00 00 00 00 00 00 ff  d8
4660 : 0c cf 3c c0 00 00 00 66  48
4668 : 66 ff 66 ff 66 66 00 31  63
4670 : fc 1e 0f e3 00 c7 31 8c  d8
4678 : 63 38 c0 79 99 e3 8c f9  35
4680 : 9f 80 36 c0 00 00 36 cc  77
4688 : c6 30 c6 33 36 c0 00 66  b5
4690 : 3c ff 3c 66 00 00 00 c3  2f
4698 : 3f 30 c0 00 00 00 de 00  9b
46a0 : 00 3f 00 00 00 00 3c 00  31
46a8 : 0c 30 c3 0c 30 00 7b 3d  aa
46b0 : fb cf 37 80 30 c7 0c 30  43
46b8 : cf c0 7b 30 c6 63 0f c0  11
46c0 : 7b 30 ce 0f 37 80 0c 38  01
46c8 : f6 6f e1 83 00 ff 0f 83  a2
46d0 : 0f 37 80 7b 3c 3e cf 37  6e
46d8 : 80 ff 31 8c 30 c3 00 7b  4e
46e0 : 3c de cf 37 80 7b 3c df  fb
46e8 : 0f 37 80 0c 30 01 80 de  ff
46f0 : 1c c6 30 60 c1 c0 00 0f  c8
46f8 : c0 fc 00 00 e0 c1 83 18  91
4700 : ce 00 7b 30 c6 30 03 00  ad
4708 : cc 07 83 7f 37 c0 31 ec  41
4710 : ff cf 3c c0 fb 3c fe cf  5b
4718 : 3f 80 7b 3c 30 c3 37 80  fd
4720 : f3 6c f3 cf 6f 00 ff 0c  4f
4728 : 3c c3 0f c0 ff 0c 3c c3  fb
4730 : 0c 00 7b 3c 37 cf 37 80  73
4738 : cf 3c ff cf 3c c0 f6 66  92
4740 : 66 f0 3c 61 86 1b 67 00  38
4748 : cf 6f 38 f3 6c c0 c3 0c  4f
4750 : 30 c3 0f c0 c7 df fe bc  2f
4758 : 78 f1 80 cf bf ff df 3c  d7
4760 : c0 7b 3c f3 cf 37 80 fb  1c
4768 : 3c fe c3 0c 00 7b 3c f3  4a
4770 : cd e1 c0 fb 3c fe f3 6c  42
4778 : c0 7b 3c 1e 0f 37 80 fc  6f
4780 : c3 0c 30 c3 00 cf 3c f3  25
4788 : cf 37 80 cf 3c f3 cd e3  6f
4790 : 00 c7 8f 1e bf fd f1 80  d0
4798 : cf 37 8c 7b 3c c0 cf 3c  17
47a0 : de 30 c3 00 fc 31 8c 63  d9
47a8 : 0f c0 cc 0c f3 cf 37 c0  e8
47b0 : 00 00 00 00 00 00 cc 07  f2
47b8 : b3 cf 37 80 00 00 00 ff  31
47c0 : 66 66 66 00 cc c7 b3 ff  cd
47c8 : 3c c0 ff ff ff ff 00 00  64
```

Listing 6. Die Schrift »Commodore« geben Sie bitte ebenfalls mit dem MSE (Seite 159) ein

```
100 REM OLIVER STILLER                         <154>
110 REM                                        <172>
120 IF L=0 THEN POKE 56,16:CLR:L=1:LOAD "A
    LLWAND.C000",8,1                           <029>
130 DIM NA$(144):C3$=CHR$(34)                  <107>
140 PRINT"{2DOWN}AMICA WANDLER V1.0/F"         <145>
150 PRINT"{DOWN}DIESES PROGRAMM WANDELT AL
    LE STANDARD{3SPACE}";                      <055>
160 PRINT"BILDER AUF EINER DISK IN DAS AMI
    CA PAINT";                                 <091>
170 PRINT"FORMAT UM."                          <118>
171 PRINT"DIE DISKETTE DARF NUR BILDER ENT
    HALTEN"                                    <026>
180 PRINT"{2DOWN}BITTE DISK EINLEGEN UND T
    ASTE DRUECKEN":WAIT 198,1:GET A$           <110>
181 PRINT"{DOWN}BITTE WARTEN..."               <192>
190 :                                          <166>
200 FOR T=4096+8000 TO 4096+8999:POKE T,15
    2:NEXT                                     <099>
210 FOR T=4096+9000 TO 4096+9999:POKE T,10
    :NEXT                                      <133>
220 POKE 4096+10000,0                          <077>
230 :                                          <206>
240 GOSUB 430                                  <000>
250 :                                          <226>
260 IF PO=0 THEN GOTO 180                      <043>
270 :                                          <248>
280 FOR T=0 TO PO-1                            <174>
290 PRINT"{DOWN}LADE BILD:";C3$;NA$(T);C3$     <069>
300 SYS 49152,8,NA$(T)                         <027>
310 FOR X=4096+8000 TO 4096+8192:POKE X,15
    2:NEXT                                     <127>
320 PRINT"PACKE BILD ..."                      <120>
330 SYS 49152+6                                <156>
340 PRINT"LOESCHE ALTES BILD ..."              <024>
350 CLOSE 1:OPEN 1,8,15,"S:"+NA$(T):CLOSE
    1                                          <064>
360 PRINT"SPEICHERE NEUES BILD ..."            <069>
370 SYS 49152+3,8,"[B]"+LEFT$(NA$(T)+"{16S
    PACE}",13)                                 <193>
380 NEXT T                                     <042>
390 END                                        <138>
400 REM                                        <208>
410 REM DIRECTORY ZEIGEN                       <244>
420 REM                                        <228>
430 CLOSE 1:OPEN 1,8,0,"$":GET#1,A$,A$:PO=
    0                                          <130>
440 PRINT"{DOWN}BILDER DER DISKETTE:"          <186>
450 FOR T=0 TO 7:GET#1,A$,A$,A$,A$:NEXT        <167>
460 :                                          <182>
470 GET#1,A$,A$,A$,A$:IF ST THEN 500           <129>
480 GET#1,X$:IF X$=C3$ THEN AN$="":GOTO 51
    0                                          <084>
490 IF ST=0 THEN 480                           <146>
500 CLOSE 1:RETURN                             <117>
510 GET#1,Z$:IF Z$<>C3$ THEN AN$=AN$+Z$:GO
    TO 510                                     <121>
520 PRINT C3$;AN$;C3$:NA$(PO)=AN$:PO=PO+1      <218>
530 GET#1,X$:IF X$<>"" AND ST=0 THEN 530       <154>
540 GOTO 470                                   <096>
```

Listing 7.
Wollen Sie Farbbilder in das Amica-Paint-Format wandeln, benötigen Sie das Programm »Allwand/F«. Es wird mit dem Checksummer (Seite 159) eingegeben.

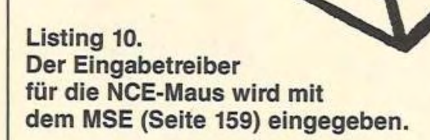
```
Name : allwand.c000        c000 c0da
-----------------------------------
c000 : 4c b4 c0 4c c6 c0 a9 10   99
c008 : 85 ff a9 37 85 fb a9 40   3d
c010 : 85 fd a9 00 85 fe 85 fc   5f
c018 : a9 11 85 fa c6 fa a0 ff   d1
c020 : b1 fa e6 fa c8 49 01 91   65
c028 : fa a0 00 20 68 c0 88 c8   b7
c030 : c0 ff f0 05 d1 fe f0 f7   95
c038 : c8 c0 03 b0 0f c9 c2 d0   23
c040 : 05 a0 01 4c 4c c0 20 7f   a9
c048 : c0 4c 29 c0 48 a9 c2 20   ae
c050 : 7f c0 98 20 7f c0 68 20   39
c058 : 7f c0 88 98 18 65 fe 85   20
c060 : fe 90 c6 e6 ff 4c 29 c0   bd
c068 : b1 fe e6 fe d0 02 e6 ff   ea
c070 : a6 ff e4 fb 90 08 a6 fe   b0
c078 : e4 fa f0 02 b0 0c 60 a2   88
c080 : 00 81 fc e6 fc d0 02 e6   89
c088 : fd 60 68 68 a9 c2 20 7f   0d
c090 : c0 a9 00 20 7f c0 60 20   e9
c098 : fd ae 20 9e b7 a9 00 a0   d2
c0a0 : 00 20 ba ff 20 fd ae 20   4c
c0a8 : 9e ad 20 a3 b6 a6 22 a4   0c
c0b0 : 23 4c bd ff 20 97 c0 a2   70
c0b8 : 00 a0 10 a9 00 20 d5 ff   9a
c0c0 : 90 03 4c f9 e0 60 20 97   e5
c0c8 : c0 a6 fc a4 fd a9 40 85   e8
c0d0 : ff a9 00 85 fe a9 fe 4c   26
c0d8 : d8 ff ff ff ff ff 00 00   b0
```

Listing 8. Diese Maschinenroutine wird von Listing 7 nachgeladen. Bitte mit dem MSE (Seite 159) eingeben.

```
100 REM OLIVER STILLER                         <154>
110 REM                                        <172>
120 IF L=0 THEN L=1:LOAD "ALLWAND.C000",8,
    1                                          <210>
130 DIM NA$(144):C3$=CHR$(34)                  <107>
140 PRINT"{2DOWN}AMICA WANDLER V1.0"           <009>
150 PRINT"{DOWN}DIESES PROGRAMM WANDELT AL
    LE KOALAPAINT-";                           <159>
160 PRINT"BILDER AUF EINER DISK IN DAS AMI
    CA PAINT";                                 <091>
170 PRINT"FORMAT UM."                          <118>
180 PRINT"{2DOWN}BITTE DISK EINLEGEN UND T
    ASTE DRUECKEN":WAIT 198,1:GET A$           <110>
190 :                                          <166>
200 GOSUB 390                                  <048>
210 :                                          <186>
220 IF PO=0 THEN GOTO 180                      <001>
230 :                                          <206>
240 FOR T=0 TO PO-1                            <132>
250 PRINT"{DOWN}LADE BILD:";C3$;NA$(T);C3$     <027>
260 SYS 49152,8,NA$(T)                         <243>
270 PRINT"PACKE BILD ..."                      <070>
280 SYS 49152+6                                <106>
290 PRINT"LOESCHE ALTES BILD ..."              <230>
300 CLOSE 1:OPEN 1,8,15,"S:"+NA$(T):CLOSE
    1                                          <014>
310 PRINT"SPEICHERE NEUES BILD ..."            <019>
320 SYS 49152+3,8,"[B]"+MID$(NA$(T)+"{16SP
    ACE}",6,16)                                <076>
330 NEXT T                                     <248>
340 END                                        <088>
360 REM                                        <168>
370 REM DIRECTORY ZEIGEN                       <204>
380 REM                                        <188>
390 CLOSE 1:OPEN 1,8,0,"$:{ORANGE}PIC *":G
    ET#1,A$,A$:PO=0                            <202>
400 PRINT"{DOWN}BILDER DER DISKETTE:"          <146>
410 FOR T=0 TO 7:GET#1,A$,A$,A$,A$:NEXT        <127>
420 :                                          <142>
430 GET#1,A$,A$,A$,A$:IF ST THEN 460           <028>
440 GET#1,X$:IF X$=C3$ THEN AN$="":GOTO 47
    0                                          <066>
450 IF ST=0 THEN 440                           <098>
460 CLOSE 1:RETURN                             <077>
470 GET#1,Z$:IF Z$<>C3$ THEN AN$=AN$+Z$:GO
    TO 470                                     <020>
480 PRINT C3$;AN$;C3$:NA$(PO)=AN$:PO=PO+1      <176>
490 GET#1,X$:IF X$<>"" AND ST=0 THEN 490       <051>
500 GOTO 430                                   <246>
```

Listing 9. »ALLWAND« wandelt Bilder vom Koalapainter nach Amica-Paint

```
Name : input.maus nce      cf00 cfe9
-----------------------------------
cf00 : 4c 60 cf a2 0e a0 cf 8e  07
cf08 : 00 c1 8c 01 c1 60 a6 e1  aa
cf10 : a4 e2 a9 a0 85 d3 a9 c8  d3
cf18 : 20 61 cf 86 e1 84 e2 08  8b
cf20 : 20 18 c0 ad 00 dc 29 10  de
cf28 : f0 25 28 a9 00 2a f0 18  2f
cf30 : ae 8d 02 e0 04 90 02 85  19
cf38 : b3 a6 b3 d0 04 a6 ee d0  18
cf40 : 02 85 fb 85 ee 4c 31 ea  a0
cf48 : 85 b3 85 ee 4c 31 ea a9  33
cf50 : 01 85 c6 a9 03 8d 77 02  79
cf58 : a9 3f 85 cb 68 4c 7e ea  34
cf60 : 78 85 b2 86 b0 84 b1 a9  61
cf68 : 10 8d 02 dc a2 08 20 b0  a7
cf70 : cf 18 65 b0 85 b0 a2 05  2d
cf78 : 20 b0 cf 18 65 b1 20 a9  9f
cf80 : cf a8 c5 b2 90 03 a4 b2  84
cf88 : 88 a5 b0 20 a9 cf aa c5  62
cf90 : d3 90 03 a6 d3 ca a9 ff  7b
cf98 : 8d 02 dc a9 7f 8d 00 dc  b1
cfa0 : ad 19 d4 c9 ff a9 01 58  4a
cfa8 : 60 c9 e3 90 02 a9 00 60  26
cfb0 : ad 00 dc 29 ef 8d 00 dc  df
cfb8 : 20 e2 cf ad 00 dc 0a 0a  16
cfc0 : 0a 0a 85 9e ad 00 dc 09  65
cfc8 : 10 8d 00 dc a2 05 20 e2  d3
cfd0 : cf ad 00 dc 29 0f 05 9e  6e
cfd8 : 49 ff 85 9e a9 00 38 e5  9d
cfe0 : 9e 60 ea ea ea ca d0 fa  04
cfe8 : 60 00 ff ff ff ff 00 00  48
```



Listing 10. Der Eingabetreiber für die NCE-Maus wird mit dem MSE (Seite 159) eingegeben.

```
Name : input.maus lms      cf00 d000
-----------------------------------
cf00 : 4c 68 cf a2 18 a0 cf 8e  ab
cf08 : 00 c1 8c 01 c1 a2 e5 a0  36
cf10 : cf 8e 8f 02 8c 90 02 60  61
cf18 : a6 e1 a4 e2 a9 a0 85 d3  92
cf20 : a9 c8 20 70 cf 86 e1 84  05
cf28 : e2 08 20 18 c0 ad 01 dc  51
cf30 : 4a 90 25 28 a9 00 2a f0  36
cf38 : 18 ae 8d 02 e0 04 90 02  bf
cf40 : 85 b3 a6 b3 d0 04 a6 ee  64
cf48 : d0 02 85 fb 85 ee 4c 81  fe
cf50 : ea 85 b3 85 ee 4c 31 ea  86
cf58 : a9 01 85 c6 a9 03 8d 77  94
cf60 : 02 a9 3f 85 cb 4c f5 cf  4e
cf68 : 78 48 18 69 01 d0 fb 68  8f
cf70 : 85 b2 86 b0 84 b1 a9 ff  82
cf78 : 8d 00 dc ad 19 d4 4a 4a  e8
cf80 : a4 9e 20 c4 cf 84 9e 18  e0
cf88 : 65 b0 20 bf cf aa c5 d3  56
cf90 : 90 03 a6 d3 ca ad 1a d4  f2
cf98 : 4a 4a a4 9f 20 c4 cf 84  95
cfa0 : 9f 38 49 ff 65 b1 20 bf  91
cfa8 : cf a8 c5 b2 90 03 a4 b2  ac
cfb0 : 88 ad 01 dc 49 ff 4a 4a  3d
cfb8 : 4a 4a 4a a9 01 58 60 c9  d7
cfc0 : e3 b0 3a 60 84 a4 c9 ff  2b
cfc8 : f0 33 85 a3 38 e5 a4 29  bf
cfd0 : 1f c9 10 b0 06 4a f0 25  af
cfd8 : a4 a3 60 09 f0 c9 ff f0  c6
cfe0 : 1c d0 16 ea ea a9 ff 8d  5e
cfe8 : 00 dc ad 01 dc 49 ff d0  9c
cff0 : 03 4c 48 eb 68 68 4c 7e  a1
cff8 : ea 38 6a 30 db a9 00 60  6b
```


Listing 11. Besitzen Sie eine Commodore-Maus, werden Sie sich für diesen Treiber interessieren. Er kann mit dem MSE (Seite 159) eingegeben werden.

```
Name : [w]maus schnell     4000 401a
-----------------------------------
4000 : 20 12 70 ee d2 00 a9 ea  cd
4008 : 8d 7e cf 8d 98 cf 8d 7f  b7
4010 : cf 8d 99 cf 20 39 a9 4c  11
4018 : 14 40 b0 e9 4a 90 10 0a  33
```

Listing 12. Der Treiber für die niedrige Geschwindigkeit der 1351-Maus. Bitte geben Sie diese Erweiterung mit dem MSE (Seite 159) ein.

```
Name : [w]maus normal      4000 401c
-----------------------------------
4000 : 20 12 70 ee d2 00 a9 4a  8c
4008 : 8d 7e cf 8d 98 cf a9 ea  fe
4010 : 8d 7f cf 8d 99 cf 20 39  0d
4018 : a9 4c 16 40 4a 90 10 0a  f2
```

Listing 13. Der Treiber für die mittlere Geschwindigkeit der 1351-Maus. Bitte geben Sie diese Erweiterung mit dem MSE (Seite 159) ein.

```
Name : [w]maus langsam     4000 401a
-----------------------------------
4000 : 20 12 70 ee d2 00 a9 4a  8c
4008 : 8d 7e cf 8d 98 cf 8d 7f  b7
4010 : cf 8d 99 cf 20 39 a9 4c  11
4018 : 14 40 16 40 4a 90 10 0a  57
```

Listing 14. Der Treiber für die hohe Original-Geschwindigkeit der 1351-Maus. Bitte geben Sie diese Erweiterung mit dem MSE (Seite 159) ein.

```
100 REM EINGABETREIBER INSTALLATIONS           <043>
110 REM PROGRAMM FUER AMICA PAINT              <153>
120 REM                                        <182>
130 REM WEITERE EINGABEGERAETE KOENNEN         <162>
140 REM IN DEN DATA-ZEILEN HINZUGEFUEGT        <254>
150 REM WERDEN (MAX.10)                        <147>
155 REM                                        <217>
157 REM PROGRAMM UEBERARBEITET VON             <019>
158 REM NIKOLAUS HEUSLER 200789                <123>
190 :                                          <166>
200 READ E$(DA),L$(DA)                         <201>
210 IF E$(DA)<>"@@@"THEN DA=DA+1:GOTO 200      <198>
220 :                                          <196>
222 PRINT"{CLR,CTRL-H,DOWN}AMICA-PAINT: IN
    STALLATIONSPROGRAMM                        <089>
224 PRINT"{13SPACE}FUER EINGABEGERAETE         <126>
226 PRINT"{DOWN}NEUE VERSION VON O.STILLER
    /N.HEUSLER                                 <251>
228 PRINT"(C) MARKT & TECHNIK, SONDERHEFT
    45                                         <041>
230 PRINT"{DOWN}VERFUEGBARE EINGABEGERAETE
    :{DOWN}                                    <040>
240 :                                          <216>
250 FOR T=.TO DA-1                             <057>
260 PRINT"{2SPACE}"CHR$(65+T)" - "E$(T)        <223>
270 NEXT                                       <026>
280 :                                          <002>
290 PRINT"{DOWN}GEWUENSCHTES GERAET ?{19SP
    ACE}TTTTTTTTTTTTTTTTTTTTT{DOWN}            <023>
300 POKE 198,.                                 <209>
310 GET A$:IF A$=""THEN 310                    <057>
311 IF ASC(A$)<65 OR ASC(A$)>DA+64 THEN 31
    0                                          <123>
320 PRINT"INSTALLATION LAEUFT. BITTE WARTE
    N...{DOWN}                                 <140>
330 OPEN 1,8,15,"I0                            <033>
340 PRINT#1,"S0:A.PAINT INPT                   <235>
350 PRINT#1,"C0:A.PAINT INPT=INPUT."+L$(AS
    C(A$)-65)                                  <081>
355 GET#1,A$:PRINT A$;:IF A$<>CHR$(13)THEN
     355                                       <046>
360 CLOSE 1:PRINT"{DOWN}FERTIG.                <029>
380 :                                          <102>
390 DATA JOYSTICK/TASTATUR,STICK               <115>
400 DATA COMMODORE MAUS 1351,MAUS              <076>
410 DATA MAUS 1351 MULTISPEED,MAUS LMS         <089>
420 DATA NCE-MAUS MIT ADAPTER,MAUS NCE         <192>
430 DATA@@@,@@@                                <183>
```

Listing 15. Die neue Version des Programmes »Eingabegeräte« installiert die Treiber in den Listings 10 und 11. Bitte geben Sie dieses Programm mit dem Checksummer (Seite 159) ein, und beachten Sie die Hinweise im Artikel.

JETZT WIRD'S BUNT

Finden Sie Gefallen an kleinen, farbigen Hardcopies, und haben Sie etwas Zeit? Holen Sie Ihren 1520-Plotter aus seiner staubigen Ecke, und versorgen Sie ihn mit »Kulis« und Papier.

Wenn Sie jetzt unser kleines Programm starten, werden Sie jedes Bild, das Sie im »Koala-Painter«-Format auf Diskette gespeichert haben, auf Ihrem kleinen Liebling »ausplotten« können. Warten Sie nicht auf das Ergebnis – Ihr Plotter ist kein LQ 2500 und braucht seine Stündchen.

Sind Sie enttäuscht? Nun, auch mit der Standardausrüstung kann eine Hardcopy ganz gut aussehen. Zum Mischen aller erdenklichen Farben auf Papier werden normalerweise Cyan, Gelb, Magenta und Schwarz benötigt. Schwarz ist vorhanden, wenn auch mit Graustufen nicht sehr viel zu machen ist. Magenta und Cyan lassen sich leidlich durch Rot und Blau ersetzen. Aber Gelb? Hier hilft kein noch so genialer Trick – gelbe Farbe muß her, wenn die Farben Nummer 7, 8 und 9 beim C 64 gemischt werden sollen.

Will man sich also nicht mit der Notlösung »Hellrot« abfinden, hilft nur der Griff in den Werkzeugkasten. Wenn Sie das alles jetzt schon für verrückt halten, oder Sie ohnehin keinen Nachmittag opfern wollen, ersparen Sie sich den Rest, und wählen Sie im Auswahlmenü nach dem Laden des Bildes »1: ....«

Gelbe Farbe – aber wie?

Aha! Sie wollen also doch einen Gelb-Stift basteln! Dann sehen Sie erst einmal nach, ob Sie einen Kuli vom Typ B (siehe Bild 1) entbehren können. Wenn nicht, wird es kompliziert. Auch Typ A läßt sich umbauen, doch der Aufwand ist ungleich höher. Daher hier die Beschreibung für den Umbau von Typ B.

Trennen Sie mit einer, besser mit zwei feinen Zangen die Spitze (also das Plastik-Teil) vom Metall-Körper. Jetzt stehen Sie schon wieder vor einer Entscheidung: Wollen Sie einen »Gelb-Kuli« oder einen »Gelb-Filzstift« herstellen? Der Kuli schreibt nachher schön sauber, leider bleibt die Farbe oft recht blaß, und die Gefahr des Aussetzens ist beim »Kuli« höher. Der »Filzstift« malt ein sattes Gelb und läßt sich auch leichter mischen. Dafür schmiert er oft und schafft keine sauberen Konturen, seine Farb-Intensität kann schwanken. Einen »Kuli« können Sie nachträglich recht leicht zu einem »Filzstift« umbauen, vielleicht erleichtert Ihnen das die Wahl.

Wenn Sie sich für einen »Filzstift« entschieden haben, müssen Sie nun den Farbtransportfilz aus der Spitze herausziehen, das winzige Kügelchen wird dann schon von alleine verschwinden. Wenn Sie diesen Filz von seiner alten Farbe gereinigt (am besten mit Alkohol, sonst mit Wasser) und getrocknet haben, stecken Sie diesen wieder »vorsichtig« in die Spitze, er darf kaum von außen sichtbar sein (lieber probieren und ggf. nachschieben!). Wenn Sie einen »Kuli« basteln wollen, lassen Sie den Filz schön in seiner Spitze. Es ist etwas schwieriger, den Filz in seiner Spitze von der alten Farbe zu befreien (siehe oben), aber mit etwas Geduld...

Jetzt für beide Stift-Sorten: Holen Sie mit einer Pinzette den alten Farbspeicher aus der Metallhülse. Ersetzen Sie diesen durch einen neuen, den Sie am besten aus dem Innenleben eines gelben Filzstiftes oder aber auch aus Watte oder ähnlichem herstellen können (muß die Hülse genau ausfüllen). Jetzt ist es an der Zeit, sich gelbe Farbe zu beschaffen. Wenn Sie gelbe Tinte bekommen, seien Sie glücklich. »Senffarbene« in den handelsüblichen Patronen gibt es aber fast überall, sie ist genauso geeignet. Gleich ein Trick, der aber nicht sein muß: Sie können Patronen erst eine Weile (offen) eintrocknen lassen, die Farbsättigung kann dadurch gesteigert werden. Tränken Sie sowohl den Vorratsspeicher (in der Metallhülse) als auch den Farbtransportfilz gut in der gelben Farbe, und fügen Sie die beiden Teile wieder zusammen; Ihr Stift sollte nun aussehen wie am Anfang, Ihre Hände dürften gelb sein.

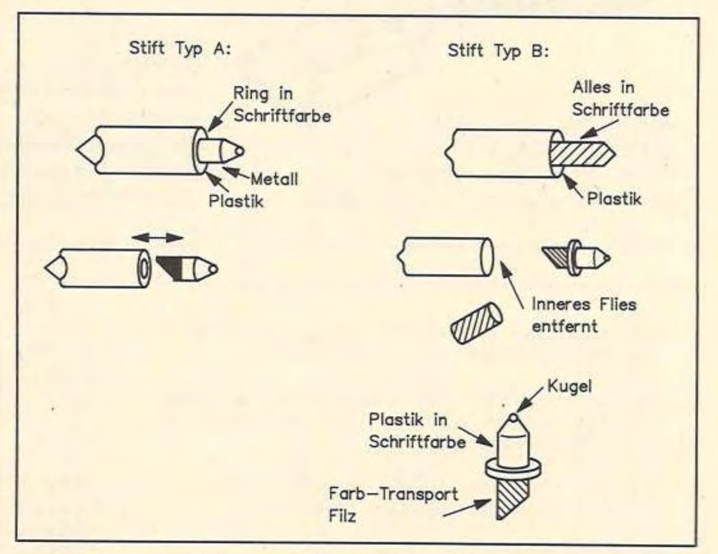


Bild 1. So entsteht ein »Gelb-Stift« für Ihren 1520-Plotter

Das Hardcopy-Programm »Col.Hardcop.1520« (Listing 1) geben Sie bitte mit dem MSE auf Seite 159 ein. Geladen und gestartet wird es wie ein normales Basic-Programm. Nach ein paar Sekunden, die das Programm zum Initiali-

Kurzinfo: Col.Hardcop.1520

Programmart: Hardcopy-Programm für den 1520-Plotter
Laden: LOAD "COL.HARDCOP.1520",8
Start: Mit RUN
Besonderheiten: Plottet alle »Koala«-Bilder in vier verschiedene Auflösungen. Nach Erstellung eines Gelb-Stiftes wird auch dieser vom Programm unterstützt.
Programmautor: Lutz Vieweg

Kurzinfo: Einschreiben

Programmart: Tool zum Einschreiben der verschiedenen Stifte des 1520-Plotters
Laden: LOAD "EINSCHREIBEN",8
Start: Mit RUN
Programmautor: Lutz Vieweg

sieren benötigt, werden Sie aufgefordert, den Namen des »Koala«-Bildes einzugeben. Wenn Sie sich nicht sicher sind, wie der Name der Grafik lautet, können Sie sich das Inhaltsverzeichnis der eingelegten Diskette mit <$> auf dem Bildschirm anzeigen lassen. Durch Druck auf eine beliebige Taste gelangen Sie wieder in das Eingabemenü. Ist das Bild im Speicher des C 64, haben Sie die Qual der Wahl unter vier verschiedenen Möglichkeiten, das Bild auf den Plotter auszugeben:

1. den vier üblichen Stiften in 320 x 200 pp (pp=Plotter-Punkte)
2. den vier Stiften + Gelb in 320 x 200 pp
3. wie oben, jedoch mit gelbem »Filz«
4. den vier Stiften + Gelb in 400 x 640 pp

Haben Sie sich für einen der vier Menüpunkte entschieden, werden Sie aufgefordert, eine Taste zu drücken. Ihr Plotter wird nun sofort mit dem Ausdruck der Hardcopy beginnen. Wie gesagt, haben Sie etwas Geduld, der Plotter ist nicht der schnellste.

Die Basic-Routine »Einschreiben« (Listing 2, bitte mit dem Checksummer auf Seite 159 eingeben) wird Ihnen helfen, den Stift zum Schreiben zu bringen, auch hier ist Geduld gefragt. Suchen Sie sich im Menü eine der beiden »Einschreib«-Formen aus. Nachdem die ersten vier Stifte ihr Werk getan haben, wird man Sie ggf. auffordern, den fünften (gelben) einzulegen. Gehen Sie dabei vorsichtig vor, um das Papier nicht zu bewegen. Die Stift-Trommel wird sich am rechten Anschlag befinden; verwenden Sie die Stift-Löse-Taste.

Hinweis:

Sollte das Programm ab und zu scheinbar »nichts« tun, rührt dies daher, daß eine Farbe lange nicht vorkommt.

Die große Hardcopy (400 x 640 Plotter-Punkte) ist bisher noch nie recht geglückt. Die großen Ungenauigkeiten des Plotters haben das Bild verschandelt. Vielleicht haben Sie mit Ihrem Plotter mehr Glück!?

Beim Ausdruck kleiner Hardcopies wird jede Farbe zweimal »geplottet«. Dies ist kein Fehler, sondern der Versuch, mehr Farbstufen möglich zu machen.

So, nun sind Sie hoffentlich gut gerüstet und können Ihre geliebten »Koala«-Bilder in Farbe auf Papier bringen.

(Lutz Vieweg/ag)

```
Name : col.hardcop.1520  0801 1223
-----------------------------------
0801 : 0f 08 50 0d 9e 32 30 35  71
0809 : 31 20 53 4f 4d 41 00 00  e8
0811 : 00 78 a2 00 86 01 bd 29  b0
0819 : 08 9d 00 04 bd b9 08 9d  75
0821 : fa 00 e8 d0 f1 4c 00 04  f9
0829 : a0 0a ca bd 23 11 9d ff  6a
0831 : fe 8a d0 f6 ce 05 04 ce  4a
0839 : 08 04 88 d0 ed ca 9a a2  64
0841 : 02 4c 00 01 06 fb f0 01  90
0849 : 60 48 a0 00 b1 fe e6 fe  a2
0851 : d0 0c e6 ff d0 08 a9 37  43
0859 : 85 01 78 4c 12 08 38 2a  9d
0861 : 85 fb 68 60 a9 00 85 a9  0e
0869 : 06 fb d0 03 20 21 04 2a  71
0871 : 26 a9 ca d0 f3 a2 02 60  56
0879 : 38 a5 fc e5 a8 85 ae a5  3c
0881 : fd e5 a9 85 af a0 00 a5  d7
0889 : fa d0 04 a4 f9 f0 28 88  5a
0891 : b1 ae 91 fc 98 d0 f8 e6  5f
0899 : af e6 fd c6 fa 10 e8 e6  b5
08a1 : fa 18 a5 f9 a8 65 fc 85  05
08a9 : fc b0 02 c6 fd 18 98 65  24
08b1 : ae 85 ae b0 02 c6 af 60  b9
08b9 : 00 80 12 08 8a f7 06 fb  f7
08c1 : d0 03 20 21 04 90 50 06  51
08c9 : fb d0 03 20 21 04 b0 04  ee
08d1 : a9 01 d0 2e 20 3c 04 c9  7c
08d9 : 03 f0 05 18 69 02 d0 22  c7
08e1 : 20 3c 04 c9 03 f0 05 18  56
08e9 : 69 05 d0 16 e8 20 3c 04  54
08f1 : c9 07 f0 05 18 69 08 d0  a9
08f9 : 09 a2 0c 20 3c 04 a4 a9  24
0901 : 84 fa 85 f9 a5 fe 85 ae  69
0909 : a5 ff 85 af 20 5d 04 a5  4d
0911 : ae 85 fe a5 af 85 ff 06  29
0919 : fb d0 03 20 21 04 b0 28  87
0921 : 06 fb d0 03 20 21 04 b0  36
0929 : 07 a2 06 20 3c 04 10 0c  43
0931 : a2 09 20 3c 04 18 69 40  0f
0939 : 90 02 e6 a9 a0 02 84 f9  d9
0941 : 85 a8 20 50 04 4c 00 01  d1
0949 : 06 fb d0 03 20 21 04 b0  5e
0951 : 36 a9 03 85 f9 20 1c 04  e6
0959 : b0 1a a2 08 a9 20 a0 00  de
0961 : 85 61 84 62 20 3c 04 18  28
0969 : 65 61 85 a8 a5 a9 65 62  f7
0971 : 85 a9 90 ce 20 1c 04 b0  1d
0979 : 06 a9 00 a2 05 d0 df a9  52
0981 : 20 a0 01 a2 0a d0 d9 06  21
0989 : fb d0 03 20 21 04 b0 09  b8
0991 : 20 1c 04 a9 00 69 04 d0  f3
0999 : ba 20 1c 04 b0 08 20 3c  2f
09a1 : 04 18 69 06 d0 ad a2 08  e2
09a9 : 20 3c 04 d0 a6 ff 82 7c  70
09b1 : 78 a2 00 86 01 bd 23 08  e6
09b9 : 9d f9 00 e8 d0 f7 4c 5c  27
09c1 : 01 15 16 01 08 ad 55 e9  0a
09c9 : c9 15 f0 33 c9 16 f0 50  71
09d1 : 91 fb e6 fb d0 02 e6 fc  4c
09d9 : ca d0 f5 e8 e6 fe d0 e5  1b
09e1 : e6 ff d0 e1 a5 fb 85 ae  e5
09e9 : 85 2d a5 fc 85 af 85 2e  56
09f1 : a2 ff 9a a9 37 85 01 58  c3
09f9 : 98 20 71 a8 4c ae a7 03  f2
0a01 : 16 fe ba ff b1 fe aa f0  e4
0a09 : 04 c9 03 90 0b 9f f5 4c  e3
0a11 : 08 01 b5 f8 a2 01 1f fe  d3
0a19 : 07 f4 a2 03 d0 ea a0 17  b8
0a21 : ca bd 4a 1e 9d 00 ff 8a  0f
0a29 : d0 f6 ce 61 01 ce 64 01  6e
0a31 : 88 d0 ed e8 4c fd 00 1b  a5
0a39 : 08 00 00 9e 32 30 37 38  07
0a41 : 20 50 4c 4f 54 54 45 52  28
0a49 : 20 48 41 52 44 43 4f 50  64
0a51 : 59 16 00 4c b4 08 68 8d  87
0a59 : 38 4f c8 39 08 4c 2f 08  42
0a61 : 20 d2 ff ee 6e 87 d0 03  34
0a69 : ee 62 ad ff 53 c0 f0 ad  4e
0a71 : 8c 48 ad 11 f9 48 60 a9  66
0a79 : 24 85 fb a9 30 7f c0 68  67
0a81 : bb a9 00 85 bc a9 01 85  ea
0a89 : b7 a9 08 85 ba a9 60 85  4d
0a91 : b9 20 d5 f3 a5 ba 20 b4  68
0a99 : ff a5 2b 2b 96 ff bf ea  d9
0aa1 : 90 a0 03 84 fb 20 a5 ff  2a
0aa9 : 85 fc a4 90 d0 2f 4a 8f  b6
0ab1 : a9 28 a4 fb 27 f1 e9 a6  0e
0ab9 : fc 20 cd bd a9 20 10 95  f7
0ac1 : 68 a6 5a 12 51 95 06 37  9d
0ac9 : ca 4c 91 08 a9 0d 47 fc  39
0ad1 : 04 09 a0 02 d0 c6 4c 42  bb
0ad9 : f6 3f 50 49 43 20 00 2a  36
0ae1 : 20 81 bd 21 08 93 0e 31  0d
0ae9 : 35 4e f8 20 c6 9b 77 42  33
0af1 : 2d c8 94 97 fc 06 02 0d  bc
0af9 : 15 12 2d 0d 0d cd 49 54  11
0b01 : 20 44 49 45 53 45 4d 20  13
0b09 : d0 52 4f 47 52 41 4d 7c  1c
0b11 : 4b 4f 45 4e 4e 03 c1 20  63
0b19 : d3 8c 0d c7 12 e0 50 48  69
0b21 : 49 4b eb 2c 00 ec a7 84  9f
0b29 : 20 49 5f b9 27 cb 4f 41  8d
0b31 : 4c 41 2d d0 41 49 4e cb  b2
0b39 : 20 27 8f 10 c6 4f 52 4d  9d
0b41 : 41 57 e9 41 55 46 20 c4  62
0b49 : 49 53 4b 20 47 09 62 50  fa
0b51 : 45 41 87 48 42 a5 c7 60  53
0b59 : 42 7e 4d 45 50 0d 99 0c  c2
0b61 : 0f bb c3 c2 cd 20 4e 17  dc
0b69 : d0 38 4b 47 46 8f 04 49  95
0b71 : 47 20 5a 55 88 b8 88 b8  eb
0b79 : 52 0d 42 52 a4 80 fc 4e  0c
0b81 : 2e 20 c7 45 83 8d 2b 72  90
0b89 : 4a 45 54 5a 0c 13 a0 89  c5
0b91 : 25 05 32 0d cb 42 62 42  44
0b99 : 55 22 cc 53 54 1f aa 7e  83
0ba1 : 64 53 20 c2 49 4c 7a 41  73
0ba9 : 26 c1 4f 24 2f ed 27 24  4f
0bb1 : 27 0d 46 55 27 05 44 41  c9
0bb9 : b1 c9 4e 24 70 4c 54 53  c8
0bc1 : 56 7d 5a 87 99 4e 33 4a  cb
0bc9 : 3a fb dc 22 22 14 12 c1  0b
0bd1 : 92 c0 af 15 3f 20 16 2e  fc
0bd9 : 4e ec 45 0d f8 22 0d 91  88
0be1 : 15 06 1d 00 58 b1 93 f5  8e
0be9 : c6 20 e4 ff f0 fb c9 24  57
0bf1 : d0 14 98 49 db 00 20 45  e3
0bf9 : 08 5b ac a5 c6 f0 fc ce  14
0c01 : fd 66 c9 41 90 da c9 5b  8a
0c09 : b0 d6 8d b2 67 20 12 c2  23
0c11 : e9 4f 01 16 91 1f a2 0c  5a
0c19 : 20 b8 09 10 83 a7 00 49  e2
0c21 : a7 a2 08 a0 73 ba 94 e2  55
0c29 : 07 a2 ad a0 aa bd cb dd  84
0c31 : a2 00 a0 6f f6 d5 e3 ce  35
0c39 : 27 ce 55 cc 47 55 82 7d  da
0c41 : 57 4f 4c 4c cb 2d 88 f6  13
0c49 : 2b 19 82 64 04 42 d9 31  4a
0c51 : 3a 8d 2d 20 34 20 51 99  5e
0c59 : 42 4c 98 6a 80 fa 54 49  f8
```

Listing 1. Das Maschinenprogramm »Col.Hardcop.1520«, bitte mit dem MSE auf Seite 159 eingeben

```
0c61 : 46 b2 41 05 cf 8c 20 33  3a
0c69 : 26 2a 03 f3 30 50 50 0d  c4
0c71 : 32 a0 43 ab d5 2b 84 07  10
0c79 : 96 f0 e0 2d 33 01 48 5e  7e
0c81 : 4f c1 e2 c8 60 f2 c7 0c  57
0c89 : 27 c6 45 b2 5a eb bc 34  1b
0c91 : 68 3f 1d 34 e0 2a 36 2a  f3
0c99 : 0f c9 0d 47 19 44 52 ba  2c
0ca1 : cf 14 c1 9f 53 bc 4e be  b0
0ca9 : 61 28 7b 3d 9c aa 94 d0  b8
0cb1 : 74 4b 0f 8a 29 0d d9 d6  f0
0cb9 : 0e ff c0 41 38 e9 31 90  d8
0cc1 : f6 c9 04 b0 f2 16 0a 18  eb
0cc9 : 69 60 85 ff 65 e2 84 fe  41
0cd1 : d1 ff 80 86 b1 fe 99 00  0c
0cd9 : 10 c8 d0 f8 e6 ff ee 66  97
0ce1 : 0b ca d0 f0 3a b3 cd 41  9e
0ce9 : 85 d3 c3 f3 33 6a 49 48  03
0cf1 : 52 ce 48 18 0d 4b 8d 65  ec
0cf9 : 52 d8 b3 14 9e e2 48 2e  a5
0d01 : 53 4f 46 4f b5 04 39 25  22
0d09 : 53 62 47 92 d4 82 2e 34  34
0d11 : b6 cc 82 cc a9 93 df 52  c3
0d19 : 27 da 15 00 00 1b 88 cf  8d
0d21 : 0c 98 f1 8b 0f 30 08 3d  74
0d29 : 80 80 f3 d0 fe 39 31 00  7f
0d31 : af f8 08 02 14 15 07 00  a5
0d39 : a5 af 16 4a a8 b9 e1 12  89
0d41 : 85 fe b9 fa fe 17 ff a5  07
0d49 : ae 0a a8 a5 ff 69 57 c2  09
0d51 : ff 98 29 f8 18 65 fe f5  9a
0d59 : 90 fc e2 5c 29 07 f0 a9  8d
0d61 : bf 86 ae 29 03 a8 aa b9  c8
0d69 : ab 12 e5 ff 01 70 31 fe  ed
0d71 : e0 00 f0 06 0a 0a ca 4c  03
0d79 : 4a 10 85 fb 24 fb 30 37  fd
0d81 : 70 06 ad 10 47 bb 60 81  39
0d89 : be 39 af 12 ba 40 1d 76  21
0d91 : c8 12 69 3f 0f 33 4a 4a  ed
0d99 : b1 30 f4 ea fc 29 f0 15  04
0da1 : 04 4a 81 b9 70 2b e4 c0  d7
0da9 : fb 0f c3 47 37 28 00 ab  12
0db1 : 43 c6 01 5f f0 0f ad 04  ca
0db9 : 10 4a ad 03 10 6a 85 ae  82
0dc1 : a9 c7 38 ed 05 4c 3e aa  1b
0dc9 : af 20 0a 10 a5 fb 05 57  0a
0dd1 : f2 29 01 f0 01 c8 84 fb  17
0dd9 : ad 09 f1 70 05 0a ee fb  ea
0de1 : 42 52 6b bd 19 ee 8c 2f  78
0de9 : 11 68 2d 58 bb 03 06 ac  ca
0df1 : cf 60 d3 90 28 8d 94 11  5b
0df9 : 9e 5a a9 a4 27 99 a9 9d  e4
0e01 : d5 df b5 1e 11 c9 0d f0  6c
0e09 : 20 c9 14 d0 0a c0 79 54  62
0e11 : e5 bf 94 88 10 e1 29 7f  c0
0e19 : c9 20 90 db 99 6c 11 fc  cd
0e21 : c8 cc 31 d0 d0 cf 8c 5f  32
0e29 : df 60 31 32 33 34 35 36  e1
0e31 : 37 38 39 30 85 4a 78 a5  b1
0e39 : 4f 62 00 8d f4 11 8e f5  69
0e41 : 37 04 8c f3 11 0d 74 d0  09
0e49 : 06 a9 30 28 e1 60 a2 ff  e1
0e51 : ad 8e 38 f9 f6 11 af c2  cf
0e59 : ad f8 f9 fb 12 06 da e8  0f
0e61 : b0 ea b5 79 d5 5b 79 6a  16
0e69 : fe 0f 8a d0 0d c0 04 f0  74
0e71 : 09 2c ff 21 30 04 a9 20  be
0e79 : d0 08 09 30 ce 6c 35 ff  bb
0e81 : 82 c0 05 d0 b9 60 16 00  b6
0e89 : 10 e8 64 0a 01 27 03 4a  52
0e91 : 75 05 d0 47 90 a9 05 20  50
0e99 : 32 d8 1a ed 4c da a8 a2  ff
0ea1 : 00 4d a9 f0 26 bd f3 c0  72
0ea9 : c9 30 f0 19 de 7b 18 c4  9d
0eb1 : 88 1d 4c 1a 12 e8 82 ca  26
0eb9 : da b7 07 ae f8 60 68 8d  56
0ec1 : 5e 12 09 e4 5f 12 4c 55  6a
0ec9 : 12 46 83 ee 74 d0 03 ee  75
0ed1 : 3b d0 ad ff ff d0 f0 ad  85
0ed9 : 8c 48 ad 11 97 48 d7 c0  b3
0ee1 : 00 04 04 03 03 01 04 5e  4a
0ee9 : 03 02 02 01 01 03 42 03  c5
0ef1 : 21 03 b0 c1 00 02 38 01  eb
0ef9 : 2d c5 f6 44 65 f0 bb 66  e9
0f01 : fc 00 15 09 04 c0 30 0c  83
0f09 : 55 ff 01 8b 28 50 78 a0  38
0f11 : c8 f0 18 40 68 90 b8 e0  0f
0f19 : 08 30 58 80 a8 d0 f8 20  95
0f21 : 48 70 98 c0 64 30 15 06  08
0f29 : bb 1b 07 02 b0 97 1a 40  25
0f31 : 80 0b 22 3a 90 ea 47 f6  72
0f39 : 20 21 22 23 25 26 27 28  47
0f41 : 2a 2b 2c 2d 2f c4 76 62  69
0f49 : 37 cd 39 3a 3b 3c 3e a2  d0
0f51 : 01 dc e1 7b 4d 4c 24 13  96
0f59 : 89 56 44 65 a5 20 45 59  ee
0f61 : e3 3e ae dc 20 95 5b 0d  e1
0f69 : 9c 7d 65 a8 a6 e3 20 cc  d6
0f71 : 44 e6 a9 08 a2 06 b1 85  c0
0f79 : 20 ba 00 59 bd de 41 c0  7b
0f81 : 15 01 44 7c f0 ba 35 02  75
0f89 : 46 02 f0 ba 39 a2 08 83  34
0f91 : cb 0d 34 2c 80 5a 49 f1  59
0f99 : 20 47 12 4d 20 c9 ce 2d  71
0fa1 : c4 ea d9 51 e1 80 15 8e  0e
0fa9 : 52 07 2f 20 00 94 ad 10  ca
0fb1 : 6f 0e 6a fc 01 8d 09 10  22
0fb9 : a9 c7 8d 92 55 f2 5c 92  7f
0fc1 : 8d e6 25 8d 08 fa e9 d9  70
0fc9 : d8 cd 97 43 d0 10 [illegible] 7e  17
0fd1 : 30 03 20 13 13 a9 ff ab  c3
0fd9 : ed 4c f8 13 42 10 10 84  7b
0fe1 : 1d 05 a0 a8 ee c1 63 70  29
0fe9 : c9 6e f7 4b c9 01 d0 ca  ce
0ff1 : be c9 40 d0 c3 e8 07 21  a0
0ff9 : ce 34 bf 22 de f0 03 4c  30
1001 : c6 0d 22 52 08 e4 4c c1  7d
1009 : 13 ee 2a 8b b9 c9 04 55  34
1011 : 96 ac db b3 f3 b4 19 42  38
1019 : 53 c3 1d 91 43 80 5f 3c  f6
1021 : 60 20 46 13 60 2b ca 0b  26
1029 : af 3f 4c 8f ff ca 1f ff  50
1031 : e5 0f 60 f2 8c 09 09 bc  c3
1039 : a2 09 02 73 e1 2c f2 75  85
1041 : 57 04 00 02 ea 01 3b 6f  5d
1049 : b2 23 2f af 27 f3 16 04  c1
1051 : 15 0c 09 ff fc 95 ef 9e  28
1059 : 4b 67 b0 92 e4 d0 33 8f  97
1061 : fd 82 0d 0d 50 4c 45 41  83
1069 : 53 45 20 49 4e ff c0 4a  0d
1071 : 52 54 20 59 45 4c 4c 4f  a7
1079 : 57 20 50 45 4e 21 0d 00  bf
1081 : a9 df 3a 24 1a c1 22 44  ee
1089 : 5b d9 9e 4c bc 63 eb 57  47
1091 : 6d ff fa d3 ff fd 69 ee  aa
1099 : 1e b5 78 06 df ac 29 19  ab
10a1 : 09 04 ff fe b2 fe df 59  e1
10a9 : bc f4 2b d6 72 38 13 ea  90
10b1 : 68 51 15 e5 7e b3 9e 02  c8
10b9 : 13 27 eb 3b 39 70 08 7a  ee
10c1 : cb c2 fc 0b 38 06 13 f2  74
10c9 : cc 1f 13 f2 5f 5b 79 00  ff
10d1 : 3d e1 c3 96 7a da 05 3c  cd
10d9 : 8b d6 eb af d6 f0 e5 af  ac
10e1 : 1d 62 cf 0b 85 ae ad 06  15
10e9 : 46 66 00 3c a9 9f 15 84  df
10f1 : ae 1b ed f8 16 0a 60 af  5a
10f9 : 14 d3 3d 9d 4b 67 be 37  53
1101 : dc 6d 82 f7 1b 46 c3 3d  a1
1109 : c4 1a df 71 bc ab 3b dc  d0
1111 : 6f 35 1a f7 1b 83 8f 71  8f
1119 : 18 af 71 02 bc af dc 7c  5b
1121 : 0b 12 8e 0c 12 bd c6 0a  99
1129 : 79 e0 87 85 fb 88 63 ab  8e
1131 : 38 f9 0d 12 f0 ad be 04  6b
1139 : f9 12 81 99 ee 20 fb 79  a2
1141 : aa b6 79 d5 75 ee 23 f7  ab
1149 : 7e e2 0b 78 bf b8 8d d0  a4
1151 : 45 7b 8d 49 be e2 4c ce  b2
1159 : bd c4 9f 1b ee 20 7b 18  d2
1161 : 64 88 19 4c 31 d7 b8 85  f9
1169 : 60 ae f5 ee 35 75 ee 30  97
1171 : 76 c5 6c be e2 0d 5b ce  5e
1179 : 0e 71 ee 20 8c 48 ad 11  e3
1181 : f0 48 60 15 08 00 0e 09  1b
1189 : 03 ea 03 01 16 09 18 0d  07
1191 : 49 83 03 e0 02 38 01 0e  7b
1199 : 04 49 8a 1b 23 59 00 1a  79
11a1 : 09 19 b7 04 e1 59 f2 60  1b
11a9 : e5 0e 02 a1 a0 a8 49 9f  fe
11b1 : 1e a1 90 65 05 ec 9f 1f  e5
11b9 : 78 7b bf c7 dd ac fc 7e  0c
11c1 : 64 ae 06 e7 a5 7c 85 5e  0c
11c9 : c7 c8 6f 36 33 39 93 fc  dc
11d1 : 80 55 7c 85 7f 9f 21 89  58
11d9 : 02 8d 31 e1 9f 2b 78 eb  37
11e1 : 77 ca dc 6a fc ae 71 55  f7
11e9 : 14 ee 57 74 8c fc ad 90  61
11f1 : d7 ca 88 d3 08 3d 21 ce  56
11f9 : 86 7c e8 19 bf 3b 9c 23  a9
1201 : 14 7c ee 78 19 14 5f ce  6b
1209 : e5 04 14 f3 a4 f8 f2 8f  71
1211 : 9d 24 c5 c2 fb 4c 14 14  25
1219 : e4 7f 3b 78 9e 67 ce e0  bd
1221 : 08 00 ff 8d 46 02 c9 00  76
```

Listing 1. (Schluß)

```
10 OPEN 1,6,1:OPEN 8,6,0                              <234>
20 PRINT"{CLR,RVSON}F{RVOFF}LAECHE /{SPACE
   ,RVSON}E{RVOFF}INSCHREIBEN ?":POKE 198,
   0:WAIT 198,1:GET A$                                <215>
30 IF A$="E"THEN 100                                  <209>
40 PRINT#8:FOR Z=0 TO 999 STEP 2                      <135>
50 PRINT#1,"M";0;-Z:PRINT#1,"D";479;-Z:PRI
   NT#1,"M";479;-Z-1:PRINT#1,"D";0;-Z-1:NE
   XT                                                 <131>
60 GOTO 40                                            <006>
100 PRINT#8:FOR Z=0 TO 999:FOR X=0 TO 472
    STEP 8                                            <089>
120 PRINT#1,"M";X;-Z:PRINT#1,"D";X+4;-Z:NE
    XT:NEXT                                           <154>
130 GOTO 100                                          <058>
```

Listing 2. Diese kleine Basic-Routine hilft, die Stifte des Plotters einzuschreiben

Der Sprite-Grabscher

Kein Spiel, kein Programm ist mehr sicher vor Ihnen! Mit unserem kleinen Hilfsprogramm holen Sie alle Sprites aus fremden Programmen und können Sie anschließend selbst weiterverwenden.

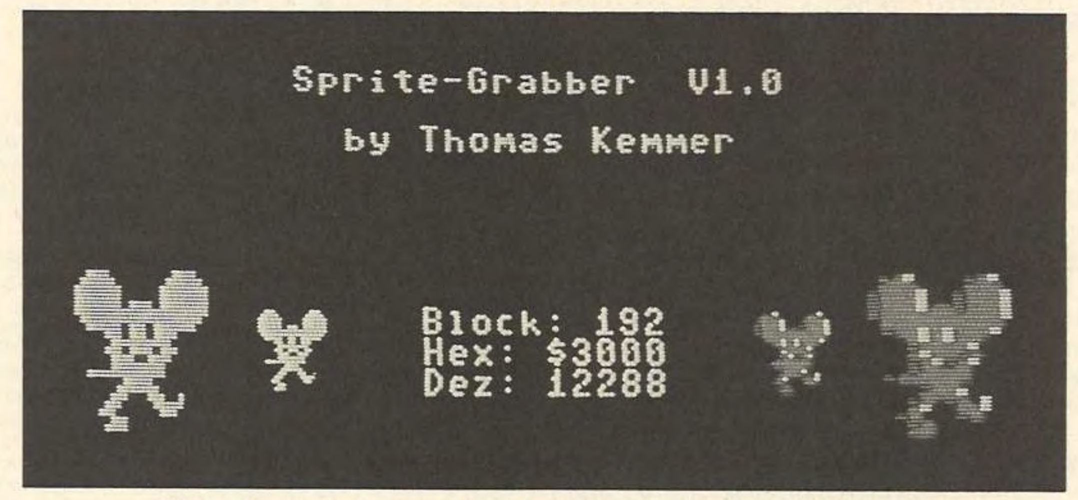


Bild 1. So erscheint das gesuchte Sprite auf dem Grabber-Bildschirm

Jeder hat wohl schon einmal die eindrucksvollen Bilder bestaunt, mit denen viele Spiele aufwarten. Und viele von Ihnen haben vielleicht schon öfters den Wunsch gehabt, die Sprites auszubauen und in eigenen Programmen zu verwenden. Natürlich nur zum Spaß und rein privat, versteht sich, weitergeben oder gar zu Geld machen dürfen Sie diese Programme dann nicht mehr.

Da das Suchen und Bearbeiten der Grafikdaten mit Hilfe eines Monitors recht unkomfortabel und umständlich ist, haben wir in letzter Zeit gelegentlich »Grafik-Diebe« veröffentlicht. Oft waren diese allerdings recht lang, weil neben den »Sprite-Grabbern« auch noch Funktionen zum »Klauen« des Grafik- oder sogar Textbildschirms enthalten waren, die man aber eigentlich nicht immer benötigt. Und dann waren Einsteiger, die die Profi-Sprites nur mal eben in ein eigenes Basic-Programm einbauen wollten, möglicherweise überfordert: Die Anzeige der Daten erfolgte im hexadezimalen Code, die Ladeadresse des erzeugten Files war nicht frei wählbar usw.

Mit unserem Listing »Sprite-Grabber V1.0« werden diese beiden Probleme gelöst. Bei der Entwicklung wurde besonders großen Wert auf einfache Bedienung und Kürze gelegt. Das Programm ist nur 969 Byte lang! Dafür eignet es sich allerdings auch »nur« zum »Entführen« von Sprites.

Das Programm (Listing 1) geben Sie zunächst mit dem MSE ein, Hinweise dazu finden Sie auf Seite 159. Speichern Sie es anschließend auf einer Diskette, auf der noch mindestens vier Blocks frei sind.

So werden Sie zum Bilderdieb

Nun eine Schritt-für-Schritt-Anleitung, wie aus einem fremden Programm Sprites »geklaut« werden:

1. Laden Sie das Programm, aus dem Sie besonders grafisch interessante Sprites »stehlen« wollen.

2. Sobald die gewünschten Figuren auf dem Bildschirm erscheinen, drücken Sie auf den Reset-Taster. Einen solchen Zusatz sollten Sie in den Computer eingebaut haben, der Einbau ist beispielsweise in Ausgabe 8/89 des 64'er Magazins beschrieben. Es gibt jedoch auch fertige Reset-Taster zu kaufen, die als kleine Platine in den User-Port des Rechners gesteckt werden. Dafür müssen Sie allerdings um die 10 Mark investieren, eine Ausgabe, die sich aber wirklich lohnt.

Sollte das Spiel einen Reset-Schutz haben und bei einem Reset neu starten, abstürzen oder den Speicher löschen, können die Sprites auf normalem Wege leider nicht entfernt werden. Sie benötigen dann ein spezielles Betriebssystem, das eine Funktion beinhaltet, die den Reset-Schutz umgeht. Bei den Systemen »Speeddos Plus« und »Exos V7« müssen Sie dazu beispielsweise gleichzeitig die Tasten <Space> und <Reset> betätigen. Übrigens: Wir reden hier immer von »Spiel«. Wie bereits erwähnt: Der Grabber kann für alle Arten von Programmen verwendet werden, die Vorgehensweise ist identisch. Nur muß es sich bei den zu klauenden Kleingrafiken unbedingt um Sprites handeln.

3. Laden Sie nun den Sprite-Grabber durch Eingabe des Befehls

Kurzinfo: Sprite-Grabber V1.0

Programmart: Tool zum »Klauen« von Sprites, die in eigenen Programmen verwendet werden können
Laden: LOAD "GRABBER",8,8
Starten: SYS 4096
Besonderheiten: Ein Reset-Taster wird benötigt
Programmautor: Thomas Kemmer

```
LOAD"GRABBER",8,8
```
und starten Sie ihn mit
```
SYS 4096
```
4. Der Bildschirm sollte sich nun schwarz färben. In der Mitte sehen Sie das gerade angewählte Sprite in vierfacher Ausführung: Single- sowie Multicolor, jeweils in normaler Größe und expandiert. Bild 1 zeigt übrigens das Sprite aus dem Listing 21 unseres Grafikkurses (Seite 116). Sie können den gesamten RAM-Bereich (auch unter den ROMs von $A000-$FFFF) nach Sprites durchsuchen. Folgende Tastenfunktionen stehen dabei zur Verfügung:

<**F1**> einen Spriteblock (64 Byte) vor
<**F3**> einen Spriteblock zurück
<**F5**> acht Sprites (512 Byte) vor
<**F7**> acht Sprites zurück
<**+**> und <**-**> byteweises Verschieben (Feineinstellung)
<**1**> bis <**3**> Spritefarben ändern
<**S**> Sprites auf Diskette speichern
<**X**> Programmende (Computer führt einen Reset aus)

Durchsuchen Sie den Speicher unter Verwendung der Funktionstasten, bis Sie etwas finden, das ein Sprite sein könnte. Dann drücken Sie die <S>-Taste. Mit den Funktionstasten, <+> und <-> legen Sie nun den zu speichernden Bereich fest und drücken <RETURN>.

5. Bestimmen Sie auf die gleiche Weise, ab welcher Adresse die Sprite-Daten später nach dem Laden im Speicher stehen sollen. Nun müssen Sie nur noch einen Filenamen (maximal 16 Zeichen) eingeben, und schon werden »Ihre« Sprites auf die Diskette geschrieben. Sie sehen, der »Sprite-Grabber« ist trotz seiner Kürze sehr einfach und trotzdem komfortabel zu bedienen. Auch Einsteiger werden mit diesem Programm keine Probleme haben, da alle Werte auch dezimal angegeben werden.

Wir wollen nicht verschweigen, daß Sie mit dem Grabber aus technischen Gründen keine Sprites »grabben« können, die in der Zeropage in den Bereichen 11 (ab 704) oder 13 bis 15 (im Kassettenpuffer, ab Speicherzelle 832) liegen. Diese Bereiche werden bei einem Reset nämlich sofort gelöscht. Das Grab-Programm steht im Speicher von 4096 bis 5065. Daraus folgt, daß die 16 Sprites, die in diesem Bereich liegen (Sprites 64 bis 79), ebenfalls nicht verfügbar sind.

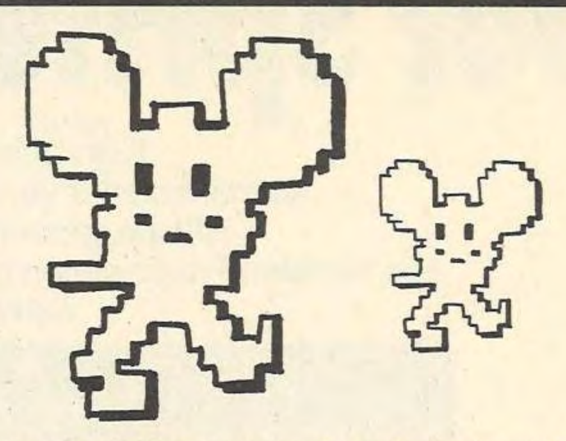
Programmtechnische Daten

Nun wollen wir auch noch einen Blick hinter die Kulissen dieses Tools werfen. Der »Sprite-Grabber V1.0« liegt im Speicherbereich von $1000 bis $13C9, einem Bereich, der nur sehr selten Grafikdaten enthält. Gewöhnlich befindet sich hier der Programmcode des Spieles. Das aktuelle Sprite wird immer im Block 13 (im Kassettenpuffer, $0340-$037f) zwischengespeichert, damit das Programm freien Zugang zu den Betriebssystem-Routinen hat. Aus Zeitgründen werden beim Speichern der Spriteblöcke jeweils 256 Byte in einem Puffer ab $1800 zwischengespeichert. Diese können Sie dann alle »in einem Aufwasch« auf Diskette schreiben. Die Startadresse, ab der die Sprites geladen werden, ist frei wählbar, so daß man die Daten mit jedem beliebigen Sprite-Editor »nachbehandeln« kann.

Das Programm zum »Grabschen« nach Sprites haben Sie von uns. Nun liegt es bei Ihnen, Ausschau nach hübschen Grafiken zu halten und diese aus den Programmen herauszuoperieren. Einer großen Sprite-Bibliothek steht nichts mehr im Wege!

(Thomas Kemmer/Nikolaus Heusler/ag)

Was ist ein Grabber?



Unter einem »Grabber« (engl. »to grab«, »grabschen«) versteht man im Computerbereich ein Hilfsmittel, das es gestattet, bestimmte Teile fremder Programme (meistens Spiele) zu isolieren und für eigene Projekte weiterzuverwenden. Dabei handelt es sich meistens um Grafik, also Hires-Bilder, Sprites, Zeichensätze. Es wären aber auch Grabber etwa für Musikstücke denkbar.

Der Grabber macht aus dem »geklauten« Objekt ein Standard-File, bei Bildern etwa ein 32 Blocks langes Hi-Eddi-File, das ganz normal geladen und weiterverwendet wird.

Bei Grabbern gibt es zwei verschiedene Arbeitsweisen. Es gibt zum einen die »Hardware-Grabber«, sie bestehen aus einer Platine oder einem Modul, das in den Expansion-Port gesteckt wird. Im abgeschalteten Zustand der Erweiterung arbeitet der Computer ganz normal. Jetzt kann beispielsweise ein Spiel geladen und gestartet werden. Wenn die gewünschte Grafik am Bildschirm erscheint, dient ein Taster am Modul dazu, dieses zu aktivieren. Das Spiel wird gestoppt. Ein im Modul eingebautes Programm erkennt die Art der Grafiken auf dem Bildschirm und erzeugt auf der Diskette automatisch ein File mit den gewünschten Grafikdaten. Diese Art des Grabbens hat den Vorteil, daß sie sehr anwenderfreundlich und sehr einfach durchzuführen ist. Auch haben Sie bei dieser Methode eigentlich höhere Erfolgschancen, da sich zum Zeitpunkt des Grabbens die Grafik ja schon/noch auf dem Bildschirm befindet. Grafiken in der Zeropage (etwa Sprites im Kassettenpuffer) können nur auf diese Art gegrabbt werden, da etwa bei einem Reset, um das Programm zu beenden, die gesamte Zeropage gelöscht wird. Dabei nullt der Computer auch die Spritebereiche aus, und die Grafikdaten gehen verloren. Beim Hardware-Grabber entfällt dieses Problem.

Vorteilhaft ist auch, daß oft nicht nur einzelne Figuren oder Objekte gegrabbt werden. Das Modul rechnet vielmehr den gesamten Bildschirm, ob er Hires-Grafik oder Text (auch mit verändertem Zeichensatz) darstellt, mit oder ohne Sprites, in eine Hires-Grafik um, die dann als Hi-Eddi-File gespeichert und sehr leicht weiterverwendet werden kann. Gelegentlich werden dabei sogar Raster-Interrupts berücksichtigt, oder es erfolgt eine Umrechnung vom Multicolor- ins Hires-Format. Einzelne Sprites können so natürlich nicht entführt werden. Der entscheidende Nachteil dieser Methode ist aber, daß Zusatzhardware erforderlich ist. Dadurch wird sie viel teurer als die reine Softwarelösung. Auch ist es ausgesprochen schwer und erfordert Fachkenntnisse, ein solches Programm selbst zu schreiben und in ein Modul einzubauen.

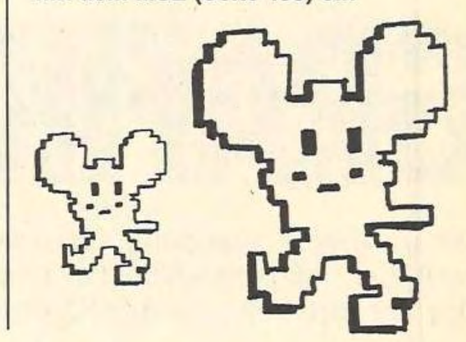
Der Grabber, den wir Ihnen anbieten, arbeitet nach einer anderen Methode. Er ist eine reine Softwarelösung, also ein einfaches Programm, das außer dem Reset-Taster keine zusätzliche Hardware erfordert. Das vorher geladene und gestartete Spiel wird durch Druck auf einen Reset-Taster abgebrochen. Dabei bleiben fast 97 Prozent des Speichers erhalten. Nun kann der Grabber von Diskette geladen und aktiviert werden. Er durchsucht nun, nach Vorgaben des Anwenders, den Speicher nach Grafik- oder Spritedaten. Die gefundenen Bereiche können wieder direkt gespeichert werden. Dem niedrigeren Preis der Software-Lösung steht der Nachteil der komplizierteren Bedienung und der etwas geringeren Erfolgschance gegenüber. Außerdem können aus den bereits erläuterten Gründen keine Zeropage-Sprites gegrabbt werden. Da auch das Grab-Programm einen kleinen Speicherbereich für sich selbst belegt, kann es in ganz ungünstigen (aber auch sehr unwahrscheinlichen) Fällen passieren, daß Sprites vom Programm überschrieben werden.

Für welche der beiden Methoden man sich letztendlich entscheidet, hängt vom Einzelfall ab. Sollten Sie nur vorhaben, Sprites zu grabben, eignet sich unser Programm sehr gut. Da es nur einen sehr kleinen Speicherplatz belegt und nach einer besonderen Technik arbeitet, kann man es praktisch in 95 Prozent aller Fälle einsetzen.

(Nikolaus Heusler/ag)

```
Name : grabber                 1000 13c9
----------------------------------------
1000 : a9 20 85 fc a9 00 8d 20   cb
1008 : d0 85 fb 8d 21 d0 a9 01   8d
1010 : a0 03 99 27 d0 88 10 fa   05
1018 : a9 02 8d 25 d0 a9 06 8d   58
1020 : 26 d0 58 a9 01 8d 86 02   94
1028 : 20 44 e5 a0 0a a2 01 18   e2
1030 : 20 f0 ff a0 14 b9 5b 13   7f
1038 : 20 d2 ff 88 10 f7 a0 0c   2e
1040 : a2 03 18 20 f0 ff a0 0f   1e
1048 : b9 70 13 20 d2 ff 88 10   72
1050 : f7 a9 20 8d 00 d0 8d 06   9e
1058 : d0 a9 80 85 02 8d 8a 02   88
1060 : a9 0d a0 03 99 f8 07 88   a7
1068 : 10 fa a9 5c 8d 02 d0 a9   6b
1070 : fc 8d 04 d0 a9 71 8d 01   ac
1078 : d0 8d 07 d0 a9 7b 8d 03   9d
1080 : d0 8d 05 d0 a9 09 8d 1d   c6
1088 : d0 8d 17 d0 a9 08 8d 10   30
1090 : d0 a9 0c 8d 1c d0 a9 0f   f7
1098 : 8d 15 d0 20 16 12 20 e7   2a
10a0 : 12 ad 36 03 c9 31 90 f3   c7
10a8 : c9 58 d0 03 20 e2 fc c9   d2
10b0 : 34 b0 17 38 e9 31 aa d0   7e
10b8 : 0b a2 03 fe 27 d0 ca 10   f9
10c0 : fa 4c 9e 10 fe 24 d0 4c   77
10c8 : 9e 10 c9 53 d0 cd a2 0e   6d
10d0 : a0 0b 18 20 f0 ff a0 11   b4
10d8 : b9 92 13 20 d2 ff a9 2d   d1
10e0 : 99 63 06 88 10 f2 a0 07   e7
10e8 : a2 11 20 34 12 a5 fb 85   eb
10f0 : fd a5 fc 85 fe a9 14 8d   58
10f8 : bb 06 a9 0f 8d bc 06 d0   7b
1100 : 14 20 e7 12 a5 fe c5 fc   c4
1108 : 90 0b d0 06 a5 fb c5 fd   60
1110 : b0 03 20 3f 13 20 16 12   e1
1118 : a2 11 a0 17 20 34 12 ad   95
1120 : 36 03 c9 0d d0 db a5 fb   66
1128 : 18 69 3f 8d 3e 03 a5 fc   03
1130 : 8d 3f 03 a0 05 b9 a8 13   19
1138 : 99 48 07 88 10 f7 20 3f   88
1140 : 13 a9 00 85 c6 20 48 13   8d
1148 : a2 15 a0 07 20 34 12 20   aa
1150 : e7 12 ad 36 03 c9 0d d0   c7
1158 : ec a2 15 a0 13 18 20 f0   43
1160 : ff a0 09 b9 ae 13 20 d2   d2
1168 : ff 88 10 f7 a9 00 85 cc   f9
1170 : 85 c6 8d 8a 02 8d 34 03   70
1178 : 20 e4 ff f0 fb ac 34 03   24
1180 : c9 0d f0 21 c9 14 f0 10   51
1188 : c0 08 f0 ec 99 c0 13 ee   f0
1190 : 34 03 20 d2 ff 4c 78 11   0f
1198 : c0 00 f0 dc ce 34 03 20   0b
11a0 : d2 ff 4c 78 11 a9 01 85   01
11a8 : cc a2 08 a0 01 20 ba ff   d7
11b0 : ad 34 03 18 69 08 a2 b8   0e
11b8 : a0 13 20 bd ff 20 c0 ff   a6
11c0 : a2 01 20 c9 ff a5 fb 20   81
11c8 : d2 ff a5 fc 20 d2 ff 78   2c
11d0 : a9 35 85 01 a0 00 b1 fd   62
11d8 : 99 00 18 c8 d0 f8 a9 37   7a
11e0 : 85 01 ad 3e 03 38 e5 fd   9e
11e8 : aa ad 3f 03 e5 fe f0 0e   cf
11f0 : b9 00 18 20 d2 ff c8 d0   a5
11f8 : f7 e6 fe 4c cf 11 b9 00   18
1200 : 18 20 d2 ff c8 ca d0 f6   f1
1208 : a9 01 20 c3 ff 20 cc ff   e6
1210 : 20 3f 13 4c 22 10 78 a9   f6
1218 : 35 85 01 a0 3e b1 fb 99   f9
1220 : 40 03 88 10 f8 a9 37 85   cb
1228 : 01 58 20 48 13 a2 09 a0   12
1230 : 0f 4c 34 12 8c 34 03 18   5b
1238 : 20 f0 ff a0 06 b9 80 13   3b
1240 : 20 d2 ff 88 10 f7 a6 02   3a
1248 : e0 64 b0 0e a9 30 20 d2   8a
1250 : ff e0 0a b0 05 a9 30 20   f7
1258 : d2 ff a9 00 20 cd bd 20   3c
1260 : c1 12 a0 05 b9 87 13 20   57
1268 : d2 ff 88 10 f7 a5 fc 20   3f
1270 : cd 12 a5 fb 20 cd 12 20   28
1278 : c1 12 a0 04 b9 8d 13 20   80
1280 : d2 ff 88 10 f7 a6 [illegible] a4   64
1288 : fc c0 00 d0 12 e0 0a b0   b0
1290 : 05 a9 20 20 d2 ff e0 64   ef
1298 : b0 05 a9 20 20 d2 ff c0   53
12a0 : 03 90 06 d0 09 e0 e8 b0   24
12a8 : 05 a9 20 20 d2 ff c0 27   0c
12b0 : 90 06 d0 09 e0 10 b0 05   f4
12b8 : a9 20 20 d2 ff 98 4c cd   65
12c0 : bd 38 20 f0 ff e8 ac 34   22
12c8 : 03 18 4c f0 ff 48 4a 4a   08
12d0 : 4a 4a 20 db 12 68 29 0f   ea
12d8 : 4c db 12 c9 0a 90 03 18   31
12e0 : 69 07 69 30 4c d2 ff 20   c9
12e8 : e4 ff f0 fb 8d 36 03 c9   b1
12f0 : 85 90 30 d0 0c a5 fb 18   f2
12f8 : 69 40 85 fb 90 13 e6 fc   99
1300 : 60 c9 86 f0 0d c9 87 f0   24
1308 : 15 c9 88 d0 da c6 fc c6   a3
1310 : fc 60 a5 fb 38 e9 40 85   04
1318 : fb b0 f6 c6 fc 60 e6 fc   6a
1320 : e6 fc 60 c9 2b f0 11 c9   e8
1328 : 2d f0 04 8d 34 03 60 a5   a8
1330 : fb d0 02 c6 fc c6 fb 60   a3
1338 : e6 fb d0 d5 e6 fc 60 a5   2e
1340 : fd 85 fb a5 fe 85 fc 60   84
1348 : a5 fc 8d 35 03 a5 fb a0   04
1350 : 06 4e 35 03 6a 88 d0 f9   4d
1358 : 85 02 60 0e 30 2e 31 d6   9f
1360 : 20 20 52 45 42 42 41 52   ad
1368 : c7 2d 45 54 49 52 50 d3   b2
1370 : 52 45 4d 4d 45 cb 20 53   3c
1378 : 41 4d 4f 48 d4 20 59 42   75
1380 : 20 3a 4b 43 4f 4c c2 24   a3
1388 : 20 3a 58 45 c8 20 3a 5a   af
1390 : 45 c4 53 4b 43 4f 4c c2   db
1398 : 2d 45 54 49 52 50 d3 20   dd
13a0 : 45 56 41 d3 0d 0f 12 46   f9
13a8 : 53 14 01 12 14 3a 20 3a   90
13b0 : 45 4d 41 4e 45 4c 49 c6   1f
13b8 : 47 52 41 42 42 45 44 2d   7b
13c0 : 20 20 20 20 20 20 20 20   c0
13c8 : 00 fc a1 fd 21 fd 21 fd   f1
```

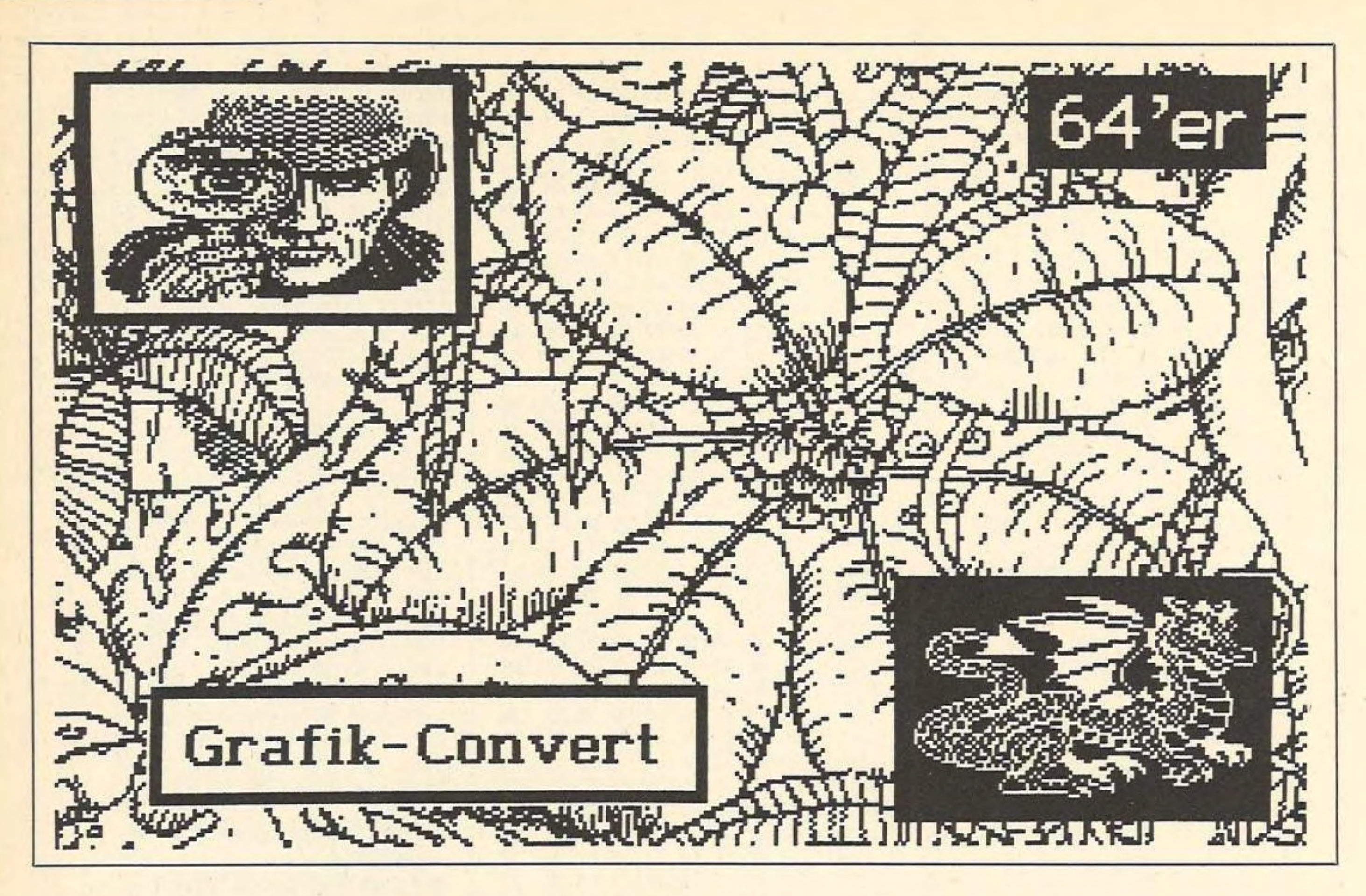
Listing 1. Bitte geben Sie den »Sprite-Klau« mit dem MSE (Seite 159) ein











GRAFIK-CONVERT 64

Mit unserer Anwendung »verstehen« sich jetzt auch die unterschiedlichsten Grafikprogramme. Das Konvertieren wird zum Kinderspiel!

Grafik-Convert ist eine Softwareschnittstelle zwischen Printshop-, Printmaster-, Starpainter-, Newsroom- und Standardgrafiken. Mit diesem Programm sind Sie ab sofort in der Lage, die in Tabelle 1 aufgeführten Grafikformate zu laden und als anderes Format aus der Tabelle wieder zu speichern.

Eine der Stärken des Programms ist das beliebige Positionieren kleiner Grafiken auf dem Hires-Bildschirm. So können beispielsweise kleinere Bilder der Newsroom »Clipart«-Disk (müssen vorher mit Newsroom als »Photo« gespeichert werden) in ein Banner oder Kalender des Printmasters eingearbeitet werden.

Die zeitkritischen Funktionen sind in reiner Maschinensprache geschrieben, wodurch das Programm einerseits sehr komfortabel zu bedienen ist und andererseits die erforderlichen Datenumwandlungen sehr schnell sind.

Art der Datei	Filename	Bemerkungen
Newsroom-Photo	PH.xxxxxx	
Newsroom-Banner	BN.xxxxxx	
Printshop-Grafik	xxxxxx	(Länge: 2 Blocks)
Printmaster-Grafik	xxxxxx.GRA	(Länge: 3 Blocks)
Standard-Grafik	xxxxxx	(es werden nie mehr als 33 Blöcke von Disk geladen)
Starpainter-HRG	xxxxxx.GR	(Filename muß 16 Zeichen lang sein)

Tabelle 1. Diese Grafikformate werden verarbeitet

Bitte geben Sie zunächst die vier Teile des Listings ein. Listing 1 und 2 stellen das Lade- und Hauptprogramm dar, bitte mit dem Checksummer eingeben. Die Listings 3 und 4 sind die dazugehörenden Maschinenprogramme und mit dem MSE einzugeben. Jetzt sollten die Programme unter folgenden Namen auf der Diskette stehen:

Listing 1	(Ladeprogramm):	Grafik-Convert
Listing 2	(Hauptprogramm):	PM-MAIN
Listing 3	(Maschinenprogramm):	PM-OBJ
Listing 4	(Sprites):	PM-SPRITE

Listing 1 erzeugt ein kleines Maschinenprogramm im Kassettenpuffer, welches ein einfaches Nachladen von Diskette ermöglicht, ohne wichtige Programmzeiger in der Zeropage zu verändern. Dieses Ladeprogramm muß immer als erstes geladen und mit RUN gestartet werden, um Grafik-Convert 64 zu aktivieren. Der wichtigste Teil von Grafik-Convert ist das Maschinenprogramm PM-OBJ. Es steht ab $8000 (32768) im Speicher und beinhaltet alle wichtigen Routinen. In PM-SPRITES sind vier Sprites gespeichert, welche die Größe des verschiebbaren Auswahlfensters auf dem Bildschirm zeigen.

Die Bedienung des Programms ist recht einfach. Sie werden ständig von Menüs, Fragen und Anweisungen begleitet. Geladen wird mit

```
LOAD "GRAFIK-CONVERT",8 (RETURN)
```

und gestartet mit RUN. Nun erscheint auf dem Bildschirm die Meldung, die in Bild 1 zu sehen ist. Nach dem Laden befinden Sie sich im Hauptmenü, welches die Punkte in Bild 2 umfaßt, auf die wir hier ein wenig näher eingehen wollen.

Mit der entsprechenden Taste (<1> bis <7>) wird die gewünschte Funktion aufgerufen.

<1> Load Picture

Es erscheint das Menü in Bild 3. Sie können nun das entsprechende Grafikformat wählen und werden anschließend aufgefordert, eine Datendiskette einzulegen, auf der sich die zu ladenden Dateien befinden müssen. Findet das Programm keine entsprechenden Files, wird wieder in das Hauptmenü verzweigt. Haben Sie als Format die Standardgrafik gewählt, so wird die Grafik unmittelbar nach Bestätigung durch <Y> in den Speicher geladen. Bei allen anderen Formaten ist die Möglichkeit gegeben, die Grafik vor dem Laden mit den Cursorsteuertasten zu positionieren. Bei den Formaten »Printshop« und »Printmaster« lassen sich bis zu 28 Files aus dem Inhaltsverzeichnis der Diskette auswählen. Ein Abbruch vor dem Ende des Directory ist mit der Taste <X> jederzeit möglich. Bei beiden Formaten wird vor dem Positionieren der Grafik angezeigt, welches File als nächstes geladen wird. Mit der Taste <N> können auch versehentlich ausgewählte Grafiken übersprungen werden.

Nach der Positionierung der Grafik kann diese durch Betätigen der RETURN-Taste normal oder durch <SHIFT-RETURN> invers geladen werden. Damit die Bilder auf dem Bildschirm so aussehen, wie sie im jeweiligen Programm dargestellt werden, müssen Printmaster- und Printshop-Grafiken mit <SHIFT-RETURN>, alle anderen dagegen nur mit <RETURN> bestätigt werden. Nachdem das File geladen wurde, kann der Ausschnitt neu positioniert werden. Es läßt sich dieselbe Grafik also mehrmals laden (außer Standard). Mit <N> wird das nächste File geladen (nur Printshop und Printmaster).

Mit <X> gelangt man in den »View«-Mode (dies erfolgt auch, wenn Sie das nächste Printshop- oder Printmaster-Bild laden wollen, obwohl schon alle ausgewählten Bilder geladen wurden). Durch nochmaliges Betätigen der Taste <X> wird wieder das Hauptmenü gezeigt.

<2> Save Picture

Auch diese Funktion zeigt ein Menü zur Auswahl des Bildformates. Nachdem das gewünschte Format ausgewählt ist, wird die komplette Grafik auf dem Bildschirm gezeigt. Haben Sie das Format »Standard« gewählt, wird durch Betätigen einer Taste der Name des Bildes erfragt und die Grafik unter diesem Namen auf Diskette gespeichert. Für alle anderen Formate können Sie den zu speichernden Ausschnitt vorher mit den Cursortasten auswählen. Ein Newsroom-Foto oder Starpainter-Bild kann zusätzlich mit den Tasten <F1> bis <F4> in der Größe verändert werden. Zum Speichern wird wieder <RETURN> oder <SHIFT-RETURN> (normal/invers) betätigt. Um später auf dem Drucker einen hellen Punkt des Bildschirms als Punkt zu sehen, werden Printshop- und Printmaster-Grafiken mit <SHIFT-RETURN>, alle anderen Formate nur mit <RETURN> bestätigt.

<3> View Picture

Die Grafik wird gezeigt, der gesamte Bildschirm kann mit <I> invertiert oder mit <SHIFT-CLR/HOME> gelöscht werden. Durch Drücken der Taste <W> erscheint ein Fenster, welches sich mit den Cursortasten verschieben und mit den Funktionstasten <F1> bis <F4> vergrößern oder verkleinern läßt. Die Befehle <I> und <SHIFT-CLR/HOME> beziehen sich jetzt nur auf die Größe des Fensters. Durch nochmaliges Drücken von <W> kommen Sie wieder in den normalen View-Mode. Mit <X> können Sie in das Hauptmenü zurückkehren.

```
Photomaker 64        Version 4.0
(W) by   J.Reinert   /   Lehrte

Please Wait while Loading...
```

Bild 1. Während des Ladens sehen Sie diese Meldung

```
========================================
   Photomaker 4.0  (W) by J.Reinert
========================================

        [1]....Load Picture
        [2]....Save Picture
        [3]....View Picture
        [4]....Source Disk 8
        [5]....Dest.- Disk 8
        [6]....Disk-Command

        [7]....Exit to basic
```

Bild 2. Das Hauptmenü von Grafik-Convert 64

```
========================================
   Photomaker 4.0  Mode: Load
========================================

        [1]...Printshop
        [2]...Newsroom Photo
        [3]...Newsroom Banner
        [4]...Printmaster
        [5]...Standard
        [6]...Starpainter

        [x]...Main Menu
```

Bild 3. Die verschiedenen Grafikformate

```
========================================
 Diskcommand: Device = Source # 8
========================================

        [1]....Scratch File(s)
        [2]....Rename  File(s)
        [3]....Validate Disk 8
        [4]....Format Datadisk 8
        [5]....Catalog Disk 8
        [6]....Change Source/Dest.

        [x]....Main Menu
```

Bild 4. Diese Funktionen bietet das Disk-Menü

Kurzinfo: Grafik-Convert 64

Programmart: Tool zur Konvertierung unterschiedlicher Grafikformate
Laden: LOAD "GRAFIK-CONVERT",8
Start: Mit RUN
Besonderheiten: Beliebiges Positionieren kleinerer Grafiken auf dem Hires-Bildschirm. Alle gängigen Formate, wie Printshop, Printmaster, Starpainter oder Newsroom werden erkannt.
Programmautor: Jürgen Reinert

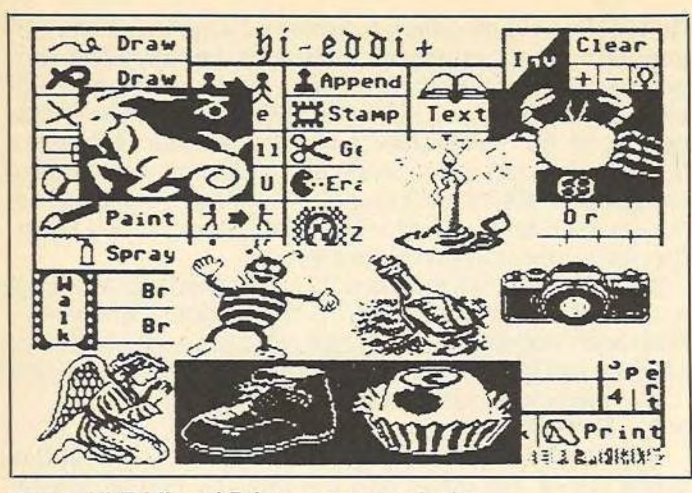


Bild 5. Hi-Eddi und Printmaster gemischt

<4> **Source Disk**
Diese Funktion schaltet die Geräteadresse bei Verwendung von zwei Laufwerken zwischen 8 auf 9 um. Das Source-Laufwerk (Source = Quelle) ist das Laufwerk, von welchem die Grafiken und »Fotos« geladen werden.
<5> **Destination Disk**
Analog zu Punkt 4 wird hiermit das Laufwerk, auf welches gespeichert werden soll, ausgewählt (Destination = Ziel).
<6> **Disk-Command**
Mit dieser Funktion gelangen Sie in das Disk-Menü (Bild 4). Hier haben Sie die Möglichkeit, Dateien auf einer Diskette komfortabel zu löschen oder umzubenennen. Weiterhin können Sie eine Diskette formatieren, einen Validate ausführen oder das Directory der Diskette einsehen.
Punkt 1: Scratch
Diskette in das geforderte Laufwerk einlegen und Taste drücken. Nun sind alle Files auszuwählen, welche auf der Diskette gelöscht (geSCRATCHt) werden sollen. Vorzeitiges Abbrechen der Auswahl erfolgt mit <X>. Nach der Sicherheitsabfrage werden alle mit <Y> bestätigten Files gelöscht.

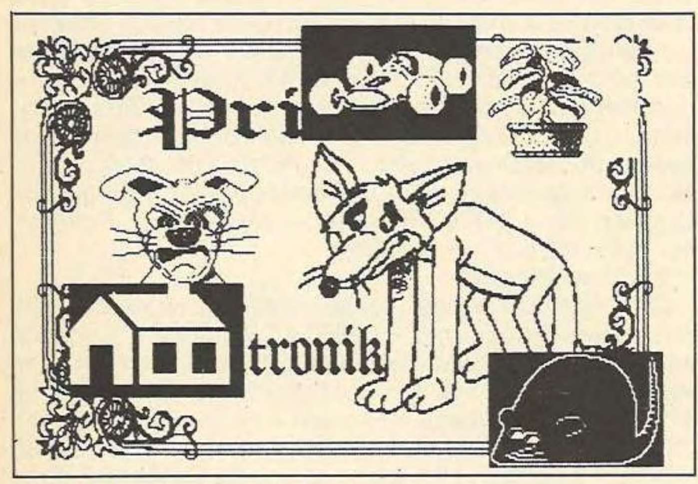


Bild 6. Der »überlagerte« Printfox

Punkt 2: Rename
Nach Einlegen der Diskette eine Taste drücken, alle Files, welche umbenannt werden sollen, durch Drücken von <Y> kennzeichnen. Anschließend werden die Files der Reihe nach umbenannt.
Punkt 3: Validate
Eine Diskette wird »aufgeräumt«, nicht geschlossene Dateien beseitigt, nicht belegter Speicherplatz freigegeben.
Punkt 4: Format Datadisk
Name und ID müssen, durch ein Komma getrennt, eingegeben werden, beispielsweise DATADISK,64. Anschließend erfolgt die obligatorische Sicherheitsabfrage. Vorsicht: Alle eventuell auf der Diskette vorhandenen Dateien sind nach dem Formatieren gelöscht!
Punkt 5: Catalog Disk
Das Inhaltsverzeichnis (Directory oder Catalog) einer Diskette wird gezeigt. Durch Betätigen einer Taste läßt sich die Anzeige stoppen. Mit <X> können Sie jetzt die Funktion abbrechen, mit einer anderen Taste wird weitergelistet.
Punkt 6: Change Source/Dest.
Wenn im Hauptmenü zwei Laufwerke eingestellt wurden (siehe Punkt 4 und 5), sind Sie in der Lage, hiermit die Befehle auf das jeweils andere Laufwerk umzuleiten. Die aktuelle Laufwerksnummer erscheint in der Kopfzeile.
Punkt 7: X
Es erfolgt ein Rücksprung in das Hauptmenü.
Soviel also zum Disk-Command-Menü. Kommen wir jetzt noch zu ein paar wichtigen Hinweisen. In das Programm wurden viele kleine, aber äußerst nützliche Routinen eingearbeitet, die den Umgang damit sehr erleichtern.
Das Programm läßt sich zu fast jedem Zeitpunkt abbrechen. Es wird dann automatisch in das Hauptmenü verzweigt. Dies geschieht auch, wenn bei einer Namenseingabe (Filename) nur die <RETURN>-Taste gedrückt wird.
Der gezeigte Ausschnitt eines Fensters (bei LOAD, SAVE und VIEW) ist so groß wie der gesamte Kasten, d.h. die Strichbreite der Sprites zählen mit zum Ausschnitt. Sobald das Fenster erscheint, kann es verschoben, bei einigen Funktionen auch in der Größe verändert werden. Dies erfolgt mit einer Schrittweite von 8 Pixel (Bildpunkten). Um das Fenster auch fein positionieren zu können, dient die Taste <S> (Steps). Mit ihr wird die Schrittweite zwischen 8 und 1 Pixel umgeschaltet (dies gilt nicht für Starpainter-Grafiken). Bei Erreichen des Bildschirmrandes wird das Fenster automatisch an den Bildschirmrand gesetzt, und zwar unabhängig davon, ob als Schrittweite 8 oder 1 Pixel gewählt wurde.

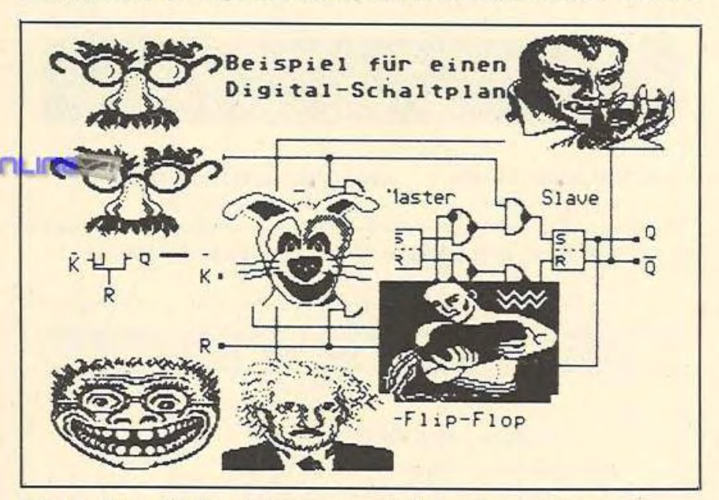


Bild 7. Eine Co-Produktion von drei Grafikprogrammen

Wenn eine Starpainter-Grafik geladen werden soll, so ist zu berücksichtigen, daß diese nicht geladen wird, wenn sie den Bildschirmrand überschreitet.
Zu bemerken ist auch noch, daß im Directory eventuell vorhandene Steuerzeichen automatisch durch Fragezeichen ersetzt werden.
Eine mögliche Anwendung von Grafik-Convert 64 ist das Verschieben eigener Grafiken in einem Newsroom-Foto: Foto nur mit <RETURN> laden, zu verschiebenden Ausschnitt speichern, Einsprung in die View-Funktion, zu verschiebenden Ausschnitt löschen, invertieren, und gespeicherten Ausschnitt an die neue Position laden — fertig!

Einen Eindruck der Leistungsfähigkeit von Grafik-Convert 64 sehen Sie in den Bildern 5 bis 7. Bild 5 zeigt ein Demo-Bild von Hi-Eddi+, welches von diversen Printmaster-Grafiken überlagert wird. In Bild 6 sehen Sie das Ladebild des Printfox, ebenfalls von kleinen und teilweise invertierten Grafikminiaturen überlagert. Bild 7 ist eine mit Grafik-Convert bearbeitete und mit dem Printfox beschriftete Hi-Eddi-Grafik. Ihrer Kreativität sind (fast) keine Grenzen mehr gesetzt.

Wenn Sie häufig mit verschiedenen Programmen arbeiten, werden Sie auf Grafik-Convert bereits nach kurzer Zeit nicht mehr verzichten wollen! (Jürgen Reinert/ag)

```
10 POKE 53280,0:POKE 53281,0:PRINT"{CLR,WH
   ITE,CTRL-H,CTRL-N,6DOWN}"                   <175>
20 POKE 55,0:POKE 56,128:CLR                   <118>
30 POKE 650,128:LO=828:I=0                     <239>
40 PRINT"{5SPACE,RVSON,30SPACE}"               <004>
50 PRINT"{5SPACE,RVSON,SPACE}PHOTOMAKER 64
   {4SPACE}VERSION 4.0 "                       <026>
60 PRINT"{5SPACE,RVSON,SPACE}(W) BY{2SPACE
   }J.REINERT{2SPACE}/{2SPACE}LEHRTE "         <008>
70 PRINT"{5SPACE,RVSON,30SPACE}"               <034>
80 READ A:IF A<>-1 THEN POKE LO+I,A:I=I+1:
   GOTO 80                                     <127>
90 DATA 032,212,225,165,185,240,005,169        <167>
100 DATA 000,076,213,255,166,043,164,044       <063>
110 DATA 169,000,032,213,255,134,045,132       <115>
120 DATA 046,032,089,166,032,051,165,169       <219>
130 DATA 000,032,144,255,032,142,166,076       <166>
140 DATA 174,167,-1                            <143>
150 PRINT"{2DOWN,6SPACE}PLEASE WAIT WHILE
    LOADING...                                 <066>
160 SYS 828"PM-OBJ",8,1                        <072>
170 SYS 828"PM-SPRITE",8,1                     <008>
180 SYS 828"PM-MAIN",8                         <226>
```

Listing 1. Das Ladeprogramm erledigt alles weitere

```
10 DIM DI$(30):ES=32768:SYS ES+51              <170>
20 XA=23:YA=23:SD=8:DD=8:LD=8:SP=8             <129>
30 REM MENUE                                   <218>
40 GOSUB 270:SYS ES+3                          <115>
50 PRINT"{CLR}";:SYS ES+6                      <062>
60 PRINT"{4SPACE}PHOTOMAKER 4.0{2SPACE}(W)
    BY J.REINERT"                              <199>
70 SYS ES+6                                    <001>
80 PRINT TAB(8);"{3DOWN}[1]....LOAD PICTUR
   E                                           <220>
90 PRINT TAB(8);"{DOWN}[2]....SAVE PICTURE     <060>
100 PRINT TAB(8);"{DOWN}[3]....VIEW PICTUR
    E                                          <169>
110 PRINT TAB(8);"{DOWN}[4]....SOURCE DISK
    ";SD                                       <082>
120 PRINT TAB(8);"{DOWN}[5]....DEST.- DISK
    ";DD                                       <255>
130 PRINT TAB(8);"{DOWN}[6]....DISK-COMMAN
    D                                          <072>
140 PRINT TAB(8);"{2DOWN}[7]....EXIT TO BA
    SIC                                        <120>
150 GET X$:ON VAL(X$)GOTO 1070,1610,2150,1
    000,1020,2230,1040                         <099>
160 GOTO 150                                   <168>
170 BL=PEEK(214)-1:IF BL<0 THEN BL=0           <211>
180 POKE 214,BL:POKE 781,BL:SYS 59903:RETU
    RN                                         <173>
190 GET#2,X$:IF X$=""THEN X$=CHR$(0)           <068>
200 XX=ASC(X$):RETURN                          <246>
210 GOSUB 170:PRINT"INSERT SOURCEDISK INTO
     DRIVE"SD:GOTO 240                         <211>
220 GOSUB 170:PRINT"INSERT DEST.-DISK INTO
     DRIVE"DD:GOTO 240                         <189>
230 GOSUB 170:PRINT"INSERT DISK INTO DRIVE
    ";LD                                       <067>
240 GOSUB 260:IF XX=88 THEN 730                <019>
250 GOSUB 170:PRINT TAB(10)"- OK -":RETURN     <006>
260 POKE 198,0:WAIT 198,1:GET X$:XX=ASC(X$
    ):RETURN                                   <125>
270 POKE 53248+21,0:RETURN                     <217>
280 OPEN 15,LD,15,"I0":GOSUB 690               <082>
290 OPEN 2,LD,2,"#":GOSUB 690:PRINT"OK."       <074>
300 T=18:S=1:N=0                               <004>
310 PRINT#15,"U1:";2;0;T;S:PRINT#15,"B-P";
    2;0                                        <030>
320 GOSUB 190:T=XX:GOSUB 190:S=XX              <059>
330 FOR X=0 TO 7:PRINT#15,"B-P";2;X*32+2       <239>
340 GOSUB 190:FT=XX                            <059>
350 IF(FT AND 128)=0 OR(FT AND 15)=0 THEN
    580                                        <173>
370 FT=FT AND 7                                <069>
380 PRINT#15,"B-P";2;X*32+5                    <248>
390 SYS ES+57,FF$                              <144>
400 PRINT#15,"B-P";2;X*32+30                   <255>
410 GOSUB 190:LB=ASC(X$):GOSUB 190:BL=ASC(
    X$)*256+LB                                 <121>
420 PRINT BL;TAB(7);FF$;TAB(25);               <211>
430 PRINT MID$("SEQPRGUSRREL",(FT-1)*3+1,3
    )                                          <253>
440 GET X$:IF M=8 THEN 510                     <149>
450 IF(M=1 AND BL=2)OR M=2 THEN 510            <234>
460 IF M=3 AND LEFT$(FF$,3)="PH."THEN 510      <126>
470 IF M=5 AND LEFT$(FF$,3)="BN."THEN 510      <196>
480 IF M=6 AND RIGHT$(FF$,4)=".GRA"THEN 51
    0                                          <172>
490 IF M=7 AND RIGHT$(FF$,3)=".GR"AND LEN(
    FF$)=16 THEN 510                           <024>
500 GOTO 580                                   <078>
510 PRINT"{UP}";TAB(33)" Y/N "                 <007>
520 GOSUB 260:IF XX=88 THEN 620                <111>
530 IF X$="N"THEN PRINT"{UP}";TAB(33)" NO{
    2SPACE}":GOTO 580                          <093>
540 IF X$<>"Y"THEN 520                         <082>
550 N=N+1:DI$(N)=FF$:IF N>27 THEN 620          <136>
560 PRINT"{UP}";TAB(33)" YES "                 <079>
570 IF M=2 OR M=3 OR M=5 OR M=7 THEN 620       <152>
580 IF X$="X"THEN 620                          <097>
590 NEXT X                                     <030>
600 IF T=0 THEN 620                            <243>
610 GOTO 310                                   <062>
620 CLOSE 2:CLOSE 15:RETURN                    <119>
630 PRINT:PRINT"{RVSON}ARE YOU SURE (Y/N)"     <184>
640 GOSUB 260:IF XX<>78 AND XX<>89 THEN 64
    0                                          <159>
650 RETURN                                     <200>
660 OPEN 15,SD,15:GOSUB 690:CLOSE 15:RETUR
    N                                          <000>
670 OPEN 2,SD,2,FF$:OPEN 15,SD,15:GOTO 690     <137>
680 OPEN 2,DD,2,F$:OPEN 15,DD,15               <075>
690 INPUT#15,A,A$,A1,A2:IF A=0 THEN RETURN     <088>
700 GOSUB 270:CLOSE 2:CLOSE 15                 <226>
710 GOSUB 170:PRINT"{RVSON}ERROR>{RVOFF}";     <206>
720 PRINT A;A$;A1;A2:SYS ES+3:GOSUB 260        <025>
730 SYS ES+48:GOTO 30                          <207>
740 IF X$="I"THEN SYS ES+12                    <160>
750 IF X$="{CLR}"THEN SYS ES+9                 <184>
760 IF M=7 THEN 800                            <058>
770 IF X$<>"S"THEN 800                         <044>
780 IF SP=8 THEN SP=1:GOTO 800                 <134>
790 SP=8                                       <029>
800 X=X+((1 AND X$="{RIGHT}")*SP)-((1 AND
    X$="{LEFT}")*SP)                           <131>
810 Y=Y+((1 AND X$="{DOWN}")*SP)-((1 AND X
    $="{UP}")*SP)                              <087>
820 IF X$="X"THEN 730                          <095>
830 IF X<0 THEN X=0                            <156>
840 IF Y<0 THEN Y=0                            <046>
850 IF X+XA>319 THEN X=319-XA                  <021>
860 IF Y+YA>199 THEN Y=199-YA                  <237>
870 RETURN                                     <166>
880 GOSUB 760                                  <206>
```

Listing 2. Das Hauptprogramm »PM-MAIN«

```
890 IF M=5 THEN 960                                  <004>
900 XA=XA+((1 AND X$="{F1}")*SP)-((1 AND X
    $="{F2}")*SP)                                    <093>
910 YA=YA+((1 AND X$="{F3}")*SP)-((1 AND X
    $="{F4}")*SP)                                    <141>
920 IF XA>MX THEN XA=MX                              <039>
930 IF XA<7 THEN XA=7                                <031>
940 IF YA>MY THEN YA=MY                              <229>
950 IF YA<7 THEN YA=7                                <060>
960 IF X+XA>319 THEN XA=319-X                        <078>
970 IF Y+YA>199 THEN YA=199-Y                        <173>
980 SYS ES+27,X,Y,XA,YA                              <147>
990 RETURN                                           <030>
1000 IF SD=8 THEN SD=9:GOTO 30                       <087>
1010 SD=8:GOTO 30                                    <126>
1020 IF DD=8 THEN DD=9:GOTO 30                       <228>
1030 DD=8:GOTO 30                                    <088>
1040 GOSUB 630                                       <056>
1050 IF X$="N"THEN 30                                <023>
1060 STOP                                            <110>
1070 LD=SD:PRINT"{CLR}";:SYS ES+6                    <075>
1080 PRINT"{4SPACE}PHOTOMAKER 4.0{2SPACE}M
     ODE: LOAD                                       <060>
1090 SYS ES+6:GOSUB 2050                             <127>
1100 ON VZ GOTO 1410,1150,1170,1420,1110,1
     160                                             <025>
1110 GOSUB 210                                       <062>
1120 M=2:PRINT"{CLR}";:GOSUB 280:IF N=0 TH
     EN 30                                           <229>
1130 SYS ES:GOSUB 670:SYS ES+21                      <053>
1140 GOSUB 690:CLOSE 15:GOTO 2150                    <087>
1150 M=3:GOTO 1180                                   <037>
1160 M=7:GOTO 1180                                   <111>
1170 M=5                                             <183>
1180 GOSUB 210                                       <132>
1190 PRINT"{CLR}";:GOSUB 280:IF N=0 THEN 3
     0                                               <171>
1200 FF$=DI$(1)                                      <209>
1210 GOSUB 670:IF M=3 OR M=5 THEN 1290               <188>
1220 GOSUB 190:XS=XX:GOSUB 190:YS=XX                 <228>
1230 CLOSE 2:CLOSE 15:IF XS>0 AND YS>0 THE
     N 1250                                          <251>
1240 GOSUB 170:PRINT"{DOWN}KEINE STARPAINT
     ER- GRAFIK":GOSUB 260:GOTO 30                   <066>
1250 IF XS<40 AND YS<25 THEN 1270                    <159>
1260 GOSUB 170:PRINT"{DOWN}GRAFIK ZU GROSS
     ":GOSUB 260:GOTO 30                             <135>
1270 XA=XS*8-1:YA=YS*8-1                             <254>
1280 X=0:Y=0:SP=8:GOTO 1320                          <208>
1290 GET#2,X$,X$,X$,X$                               <180>
1300 GOSUB 190:Y1=XX:GOSUB 190:YA=XX-Y1              <148>
1310 GOSUB 190:X1=XX-8:GOSUB 190:XA=(XX-8)
     -X1:CLOSE 2:CLOSE 15                            <102>
1320 SYS ES                                          <011>
1330 SYS ES+27,X,Y,XA,YA                             <245>
1340 GET X$:XX=ASC(X$+CHR$(0))                       <188>
1350 IF X$="X"THEN GOSUB 270:GOTO 2150               <120>
1360 IF(XX AND 127)=13 THEN F=(XX AND 128)
     /128:GOTO 1380                                  <039>
1370 GOSUB 740:GOTO 1330                             <210>
1380 GOSUB 670:IF M=7 THEN SYS ES+60,F,X/8
     ,Y/8:GOTO 1400                                  <145>
1390 SYS ES+42,F,X,Y,X+XA,Y+YA                       <186>
1400 GOSUB 690:CLOSE 15:GOTO 1330                    <012>
1410 M=1:XA=43:YA=44:EO=36:GOTO 1430                 <105>
1420 M=6:XA=87:YA=51:EO=30                           <099>
1430 GOSUB 210:PRINT"{CLR}";:GOSUB 280:IF
     N=0 THEN 30                                     <167>
1440 MX=319-XA:MY=199-MY:SYS ES+6                    <246>
1450 FOR I=1 TO N:GOSUB 270                          <126>
1460 SYS ES+3:PRINT"NEXT FILE: "DI$(I):GOS
     UB 260                                          <150>
1470 IF X$="N"THEN 1590                              <118>
1480 IF X$="X"THEN 30                                <204>
1490 SYS ES                                          <181>
1500 SYS ES+27,X,Y,XA,YA                             <159>
1510 GET X$:XX=ASC(X$+CHR$(0))                       <102>
1520 IF X$="X"THEN 1600                              <037>
1530 IF(XX AND 127)=13 THEN F=(XX AND 128)
     /128:GOTO 1560                                  <207>
1540 IF X$="N"THEN 1590                              <190>
1550 GOSUB 740:GOTO 1500                             <128>
1560 FF$=DI$(I):GOSUB 670                            <074>
1570 SYS ES+EO,F,X,Y                                 <107>
1580 GOSUB 690:CLOSE 15:GOTO 1510                    <130>
1590 NEXT I                                          <150>
1600 GOSUB 270:GOTO 2150                             <199>
1610 LD=DD:PRINT"{CLR}";:SYS ES+6                    <137>
1620 PRINT"{4SPACE}PHOTOMAKER 4.0{2SPACE}M
     ODE: SAVE                                       <045>
1630 SYS ES+6:GOSUB 2050                             <159>
1640 ON VZ GOTO 1700,1850,1860,1710,1650,1
     870                                             <227>
1650 SYS ES:GOSUB 260:PRINT:SYS ES+3                 <228>
1660 F$="":INPUT"FILENAME";F$:IF F$=""THEN
      30                                             <121>
1670 F$=LEFT$(F$,16)+",P,W"                          <201>
1680 GOSUB 220:GOSUB 680                             <070>
1690 SYS ES+24:GOSUB 690:CLOSE 15:GOTO 30            <229>
1700 M=1:XA=43:YA=44:EO=39:GOTO 1720                 <082>
1710 M=6:XA=87:YA=51:EO=33                           <071>
1720 MX=319-XA:MY=199-YA:SYS ES:GOTO 1770            <074>
1730 SYS ES+27,X,Y,XA,YA                             <135>
1740 GET X$:XX=ASC(X$+CHR$(0))                       <078>
1750 IF X$="X"THEN 30                                <220>
1760 IF(XX AND 127)=13 THEN F=(XX AND 128)
     /128:GOTO 1780                                  <189>
1770 GOSUB 740:GOTO 1730                             <108>
1780 GOSUB 270:SYS ES+3:PRINT                        <033>
1790 F$="":INPUT"FILENAME";F$:IF F$=""THEN
      30                                             <251>
1800 IF M=1 THEN F$=LEFT$(F$,15)+",P,W":GO
     TO 1820                                         <097>
1810 F$=LEFT$(F$,10)+".GRA,P,W"                      <211>
1820 GOSUB 220:GOSUB 680                             <212>
1830 SYS ES+EO,F,X,Y                                 <113>
1840 GOSUB 690:CLOSE 15:GOTO 30                      <164>
1850 M=3:MX=230:MY=167:GOTO 1880                     <203>
1860 M=5:MX=239:MY=79:XA=239:YA=79:GOTO 18
     80                                              <228>
1870 M=7:MX=319:MY=199:XA=31:YA=31:SP=8              <213>
1880 SYS ES:GOSUB 740:GOSUB 920                      <081>
1890 GET X$:XX=ASC(X$+CHR$(0))                       <230>
1900 IF(XX AND 127)=13 THEN F=(XX AND 128)
     /128:GOTO 1920                                  <065>
1910 GOSUB 880:GOTO 1890                             <092>
1920 GOSUB 270:SYS ES+3:PRINT                        <175>
1930 F$="":INPUT"FILENAME";F$:IF F$=""THEN
      30                                             <137>
1940 IF M=7 THEN F$=LEFT$(F$+"{13SPACE}",1
     3)+".GR,P,W"                                    <149>
1950 IF M=3 THEN F$=LEFT$("PH."+F$,11)+",P
     ,W"                                             <240>
1960 IF M=5 THEN F$=LEFT$("BN."+F$,11)+",P
     ,W"                                             <249>
1970 GOSUB 220:X2=X+XA:Y2=Y+YA:C=1 AND((X
     AND 7)>0)                                       <150>
1980 B=(INT(X/8)+C)*(Y+1)                            <063>
1990 GOSUB 680                                       <068>
2000 IF M=3 THEN SYS ES+45,F,X,Y,X2,Y2,B             <127>
2010 IF M=5 THEN SYS ES+54,F,X,Y,X2,Y2,B             <193>
2020 IF M=7 THEN SYS ES+63,F,X/8,Y/8,XA/8+
     1,YA/8+1                                        <150>
2030 GOSUB 690:CLOSE 15:GOTO 30                      <098>
2040 STOP                                            <072>
2050 PRINT:PRINT TAB(8);"{2DOWN}[1]...PRIN
     TSHOP                                           <045>
2060 PRINT TAB(8);"{DOWN}[2]...NEWSROOM PH
     OTO                                             <083>
2070 PRINT TAB(8);"{DOWN}[3]...NEWSROOM BA
     NNER                                            <014>
2080 PRINT TAB(8);"{DOWN}[4]...PRINTMASTER           <103>
2090 PRINT TAB(8);"{DOWN}[5]...STANDARD              <132>
2100 PRINT TAB(8);"{DOWN}[6]...STARPAINTER           <118>
2110 PRINT TAB(8);"{2DOWN}[X]...MAIN MENU            <157>
2120 GOSUB 260:IF XX=88 THEN 730                     <123>
2130 IF XX<49 OR XX>54 THEN 2120                     <116>
2140 VZ=VAL(X$):RETURN                               <130>
2150 SYS ES:M=1                                      <115>
2160 GET X$:GOSUB 740                                <227>
2170 IF X$<>"W"THEN 2160                             <094>
2180 MX=320:MY=200:GOSUB 920                         <040>
2190 GET X$:IF X$="W"THEN GOSUB 270:GOTO 2
     150                                             <083>
2200 IF X$="I"THEN SYS ES+18,X,Y,X+XA,Y+YA           <108>
2210 IF X$="{CLR}"THEN SYS ES+15,X,Y,X+XA,
     Y+YA                                            <025>
2220 GOSUB 880:GOTO 2190                             <135>
2230 PRINT"{CLR}";:SYS ES+6                          <210>
2240 PRINT"{2SPACE}DISKCOMMAND: DEVICE = "
```

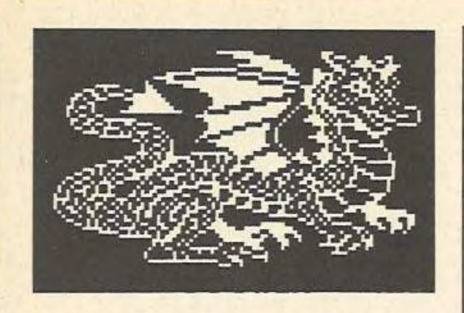
```
;                                                    <040>
2250 IF LD=SD THEN PRINT"SOURCE";:GOTO 227
     0                                               <072>
2260 PRINT"DEST. ";:LD=DD                            <218>
2270 PRINT" #"LD                                     <016>
2280 SYS ES+6:PRINT:PRINT                            <067>
2290 PRINT TAB(8)"{DOWN}[1]....SCRATCH FIL
     E(S)                                            <189>
2300 PRINT TAB(8)"{DOWN}[2]....RENAME{2SPA
     CE}FILE(S)                                      <166>
2310 PRINT TAB(8)"{DOWN}[3]....VALIDATE DI
     SK"LD                                           <129>
2320 PRINT TAB(8)"{DOWN}[4]....FORMAT DATA
     DISK"LD                                         <217>
2330 PRINT TAB(8)"{DOWN}[5]....CATALOG DIS
     K"LD                                            <061>
2340 PRINT TAB(8)"{DOWN}[6]....CHANGE SOUR
     CE/DEST.                                        <063>
2350 PRINT TAB(8)"{2DOWN}[X]....MAIN MENU            <125>
2360 GOSUB 260:IF XX=88 THEN 30                      <187>
2370 ON VAL(X$)GOTO 2470,2550,2390,2420,26
     40,2660                                         <100>
2380 GOTO 2360                                       <026>
2390 GOSUB 230:GOSUB 170:IF X$="X"THEN 223
     0                                               <245>
2400 OPEN 15,LD,15,"V0":CLOSE 15                     <058>
2410 OPEN 15,LD,15:GOSUB 690:CLOSE 15:GOTO
     2230                                            <187>
2420 GOSUB 230:GOSUB 170                             <057>
2430 INPUT"DISKNAME,ID";DN$,ID$                      <060>
2440 GOSUB 630:IF X$="N"THEN 2230                    <010>
2450 GOSUB 170:PRINT TAB(10);"PLEASE WAIT{
     UP}"                                            <201>
2460 OPEN 15,LD,15,"N:"+DN$+","+ID$:CLOSE
     15:GOTO 2410                                    <127>
2470 M=8:GOSUB 230:PRINT"{CLR}":GOSUB 280            <154>
2480 IF N=0 THEN 2230                                <237>
2490 GOSUB 630:IF X$="N"THEN 2230                    <060>
2500 SYS ES+6:OPEN 15,LD,15:FOR I=1 TO N             <109>
2510 :PRINT"SCRATCHING "DI$(I);TAB(32);              <163>
2520 PRINT#15,"S:"+DI$(I):INPUT#15,A:IF A=
     1 THEN PRINT"OK":GOTO 2540                      <024>
2530 PRINT"ERR"A                                     <247>
2540 NEXT:CLOSE 15:SYS ES+6:GOSUB 260:GOTO
     2230                                            <122>
2550 M=8:GOSUB 230:PRINT"{CLR}":GOSUB 280            <234>
2560 IF N=0 THEN 2230                                <063>
2570 GOSUB 630:IF X$="N"THEN 2230                    <142>
2580 SYS ES+6:OPEN 15,LD,15:FOR I=1 TO N             <191>
2590 PRINT DI$(I);TAB(18)"WIRD ZU";:INPUT
     F$                                              <006>
2600 PRINT#15,"R:"+F$+"="+DI$(I):INPUT#15,
     A                                               <096>
2610 IF A=0 THEN PRINT"{UP}";TAB(36)"{SPAC
     E,RVSON}OK":GOTO 2630                           <238>
2620 PRINT"{UP}";TAB(36)"{SPACE,RVSON}ERR"           <207>
2630 NEXT:CLOSE 15:SYS ES+6:GOSUB 260:GOTO
     2230                                            <214>
2640 GOSUB 230:PRINT"{CLR}";:SYS ES+66,LD            <115>
2650 OPEN 15,LD,15:GOSUB 690:CLOSE 15:GOTO
     2230                                            <173>
2660 IF LD=DD THEN LD=SD:GOTO 2230                   <172>
2670 LD=DD:GOTO 2230                                 <007>
```

Listing 2. (Schluß)

Grafik-Convert

```
Name : pm-obj                8000 8820
-------------------------------------
8000 : 4c 54 81 4c 8d 81 4c c3   fe
8008 : 80 4c 66 81 4c 2b 82 4c   39
8010 : 7e 83 4c 89 83 4c 51 82   79
8018 : 4c 83 82 4c ba 82 4c bb   b8
8020 : 83 4c ec 83 4c 3b 84 4c   be
8028 : 68 84 4c b6 84 4c 25 85   06
8030 : 4c 25 81 4c 31 81 4c 15   73
8038 : 85 4c d7 80 4c bb 86 4c   3f
8040 : e6 86 4c 45 80 a9 24 85   16
8048 : fd a9 fd 85 bb a9 00 85   5e
8050 : bc a9 01 85 b7 20 70 83   17
8058 : 86 ba a9 60 85 b9 20 d5   04
8060 : f3 a5 ba 20 b4 ff a5 b9   2e
8068 : 20 96 ff a9 00 85 90 a0   b8
8070 : 03 84 fd 20 a5 ff 85 fe   a7
8078 : a4 90 d0 3e 20 a5 ff a4   d9
8080 : 90 d0 37 a4 fd 88 d0 e9   16
8088 : a6 fe 20 cd bd a9 20 20   59
8090 : d2 ff 20 a5 ff a6 90 d0   38
8098 : 21 aa f0 06 20 d2 ff 4c   3c
80a0 : 92 80 a9 0d 20 d2 ff 20   57
80a8 : e4 ff aa f0 09 20 e4 ff   7a
80b0 : f0 fb c9 58 f0 04 a0 02   d1
80b8 : d0 b7 20 42 f6 20 e4 ff   b8
80c0 : f0 fb 60 a9 20 20 d2 ff   49
80c8 : a2 25 a9 3d 20 d2 ff ca   3d
80d0 : 10 fa a9 0d 4c d2 ff a2   0a
80d8 : 02 20 c6 ff a2 00 20 cf   e6
80e0 : ff c9 a0 f0 14 48 29 7f   31
80e8 : c9 20 b0 04 68 a9 3f 48   cf
80f0 : 68 9d 1d 88 e8 e0 10 d0   f7
80f8 : e5 8a 48 20 cc ff 20 fd   82
8100 : ae 20 8b b0 85 49 84 4a   00
8108 : 20 a3 b6 68 20 75 b4 a0   76
8110 : 02 b9 61 00 91 49 88 10   ed
8118 : f8 c8 b9 1d 88 91 62 c8   b7
8120 : c4 61 d0 f6 60 [illegible] [illegible] 68   64
8128 : a2 fa 9a 48 98 48 4c ae   51
8130 : a7 78 a9 3e 8d 14 03 a9   1e
8138 : 81 8d 15 03 58 60 a2 07   47
8140 : ee b0 87 ad b0 87 29 03   10
8148 : 8d b0 87 9d 27 d0 ca 10   07
8150 : fa 4c 31 ea a9 3b 8d 11   e7
8158 : d0 a9 08 8d 18 d0 a9 00   5f
8160 : 8d 00 dd 4c 79 81 a9 e0   fa
8168 : 85 04 a9 00 85 03 a8 91   90
8170 : 03 c8 d0 fb e6 04 d0 f7   4d
8178 : 60 a9 10 a0 00 99 00 c0   13
8180 : 99 00 c1 99 00 c2 99 00   39
8188 : c3 c8 d0 f1 60 a9 1b 8d   fc
8190 : 11 d0 a9 17 8d 18 d0 a9   87
8198 : 07 8d 00 dd 60 a9 03 2c   d9
81a0 : a9 02 2c a9 01 2c a9 00   a3
81a8 : 8d ca 87 ad d2 87 85 14   d9
81b0 : ad d3 87 85 15 ae d9 87   17
81b8 : 8a 4a 4a 4a 0a a8 b9 e4   da
81c0 : 87 85 fd b9 e5 87 85 fe   6f
81c8 : 8a 29 07 18 65 fd 85 fd   04
81d0 : a5 14 29 f8 8d ae 87 18   85
81d8 : a5 fd 6d ae 87 85 fd a5   95
81e0 : fe 65 15 85 fe a5 14 29   47
81e8 : 07 49 07 85 14 78 a2 35   00
81f0 : 86 01 a0 00 b1 fd a2 37   23
81f8 : 86 01 58 a4 14 ae ca 87   9a
8200 : f0 0e e0 02 f0 18 e0 03   c9
8208 : f0 0c 39 b1 87 4c 19 82   c7
8210 : 19 b9 87 4c 19 82 59 b9   f0
8218 : 87 a0 00 91 fd 60 39 b9   5d
8220 : 87 c9 00 d0 03 18 90 f5   c5
8228 : 38 b0 f2 a9 00 85 fb a9   1a
8230 : e0 85 fc 78 a2 35 86 01   11
8238 : a0 00 b1 fb 49 ff 91 fb   97
8240 : c8 d0 f7 e6 fc a5 fc c9   cf
8248 : 00 d0 ef a2 37 86 01 58   5d
8250 : 60 a2 02 20 c6 ff a9 00   99
8258 : 85 fb a9 e0 85 fc 20 cf   c2
8260 : ff 20 cf ff 24 90 70 13   12
8268 : 20 cf ff a0 00 91 fb e6   ce
8270 : fb d0 f1 e6 fc a5 fc c9   b1
8278 : 00 d0 e9 20 cc ff a9 02   d6
8280 : 4c c3 ff a2 02 20 c9 ff   4a
8288 : a9 00 85 fb a9 e0 85 fc   c4
8290 : a9 00 20 d2 ff a9 20 20   aa
8298 : d2 ff a0 00 78 a2 35 86   10
82a0 : 01 b1 fb a2 37 86 01 58   29
82a8 : 20 d2 ff e6 fb d0 eb e6   d2
82b0 : fc a5 fc c9 00 d0 e3 4c   a6
82b8 : 7b 82 a9 00 8d 17 d0 8d   cf
82c0 : 1d d0 8d 1b d0 20 70 83   e3
82c8 : 8e d0 87 8c d1 87 20 70   ec
82d0 : 83 8e d8 87 20 70 83 8e   72
82d8 : d6 87 8c d7 87 20 70 83   d2
82e0 : 8e da 87 ad d0 87 18 69   ef
82e8 : 18 8d d4 87 ad d1 87 69   47
82f0 : 00 8d d5 87 ad d8 87 18   0d
82f8 : 69 32 8d 01 d0 8d 03 d0   25
8300 : a9 00 20 50 83 a9 02 20   89
8308 : 50 83 ad d8 87 18 6d da   45
8310 : 87 18 69 1e 8d 05 d0 8d   21
8318 : 07 d0 ad d0 87 18 6d d6   a9
8320 : 87 8d d4 87 ad d1 87 6d   f6
8328 : d7 87 8d d5 87 ee d4 87   33
8330 : d0 03 ee d5 87 a9 01 20   02
8338 : 50 83 a9 03 20 50 83 a9   fa
8340 : 0f 8d 15 d0 a2 07 bd c1   52
8348 : 87 9d f8 c3 ca 10 f7 60   22
8350 : 48 0a aa ad d4 87 9d 00   fe
```

Listing 3. Das Maschinenprogramm »PM-OBJ« wird mit dem MSE auf Seite 159 eingegeben



```
8358 : d0 68 aa ad d5 87 4a ad  db
8360 : 10 d0 90 05 1d b9 87 b0  bc
8368 : 03 3d b1 87 8d 10 d0 60  c4
8370 : 20 fd ae 20 8a ad 20 f7  c5
8378 : b7 a6 14 a4 15 60 20 9c  2a
8380 : 83 a2 00 8e cc 87 e8 d0  74
8388 : 08 20 9c 83 a2 01 8e cc  3e
8390 : 87 8e cd 87 a2 00 8e ce  c5
8398 : 87 4c c9 85 20 70 83 8e  19
83a0 : d4 87 8c d5 87 20 70 83  58
83a8 : 8e d9 87 20 70 83 8e db  1e
83b0 : 87 8c dc 87 20 70 83 8e  56
83b8 : dd 87 60 20 70 83 8e cc  6c
83c0 : 87 20 70 83 8e d4 87 8c  ab
83c8 : d5 87 20 70 83 8e d9 87  9a
83d0 : a2 02 20 c6 ff a2 07 20  c6
83d8 : cf ff ca d0 fa a9 00 8d  8c
83e0 : cb 87 8d cd 87 a9 01 8d  71
83e8 : ce 87 d0 32 20 70 83 8e  a5
83f0 : cc 87 20 70 83 8e d4 87  a5
83f8 : 8c d5 87 20 70 83 8e d9  66
8400 : 87 a2 02 20 c9 ff a2 06  90
8408 : bd 16 88 20 d2 ff ca 10  6f
8410 : f7 a9 00 8d cd 87 a9 01  4f
8418 : 8d cb 87 8d ce 87 ad d9  b2
8420 : 87 18 69 33 8d dd 87 ad  b5
8428 : d4 87 18 69 57 8d db 87  53
8430 : ad d1 87 69 00 8d dc 87  c4
8438 : 4c c9 85 20 70 83 8e cc  c5
8440 : 87 20 70 83 8e d4 87 8c  2b
8448 : d5 87 20 70 83 8e d9 87  1a
8450 : a2 02 20 c6 ff 20 cf ff  14
8458 : 20 cf ff a9 00 8d cb 87  40
8460 : 8d cd 87 8d ce 87 f0 31  b7
8468 : 20 70 83 8e cc 87 20 70  dd
8470 : 83 8e d4 87 8c d5 87 20  36
8478 : 70 83 8e d9 87 a2 02 20  5e
8480 : c9 ff a9 00 20 d2 ff a9  9f
8488 : 58 20 d2 ff a9 01 8d cb  15
8490 : 87 a9 00 8d cd 87 8d ce  8a
8498 : 87 ad d9 87 18 69 2c 8d  f6
84a0 : dd 87 ad d4 87 18 69 2b  7c
84a8 : 8d db 87 ad d5 87 69 00  fa
84b0 : 8d dc 87 4c c9 85 20 70  41
84b8 : 83 8e cc 87 20 70 83 8e  57
84c0 : d4 87 8c d5 87 20 70 83  78
84c8 : 8e d9 87 20 70 83 8e db  3e
84d0 : 87 8c dc 87 20 70 83 8e  76
84d8 : dd 87 a2 02 20 c6 ff a2  df
84e0 : 08 20 cf ff ca 10 fa c9  99
84e8 : 00 f0 15 8d de 87 a9 0d  42
84f0 : 8d df 87 20 cf ff ce df  4b
84f8 : 87 d0 f8 ce de 87 d0 ee  4a
8500 : 20 cf ff c9 ff d0 f9 a9  03
8508 : 00 8d cb 87 8d cd 87 8d  33
8510 : ce 87 4c c9 85 a9 01 8d  b3
8518 : cb 87 8d ce 87 a9 00 8d  c5
8520 : cd 87 4c 32 85 a9 01 8d  cf
8528 : cb 87 a9 00 8d cd 87 8d  a2
8530 : ce 87 20 70 83 8e cc 87  c7
8538 : 20 70 83 8e d4 87 8c d5  aa
8540 : 87 20 70 83 8e d9 87 20  7a
8548 : 70 83 8e db 87 8c dc 87  f8
8550 : 20 70 83 8e dd 87 20 70  d6
8558 : 83 8e de 87 8c df 87 a2  f6
8560 : 02 20 c9 ff a9 00 20 d2  a5
8568 : ff a9 a0 20 d2 ff ad ce  e9
8570 : 87 f0 17 a9 60 20 d2 ff  bd
8578 : a9 09 20 d2 ff a9 2b 20  42
8580 : d2 ff a9 7a 20 d2 ff 4c  3d
8588 : a5 85 ad de 87 20 d2 ff  fc
8590 : ad df 87 20 d2 ff a9 00  e6
8598 : 20 d2 ff ad dd 87 38 ed  ae
85a0 : d9 87 20 d2 ff a9 08 20  4d
85a8 : d2 ff ad db 87 38 ed d4  fc
85b0 : 87 18 69 08 20 d2 ff a9  8b
85b8 : 00 20 d2 ff a9 ff 20 d2  3e
85c0 : ff a9 00 8d ce 87 4c c9  33
85c8 : 85 ad d4 87 8d d2 87 ad  33
85d0 : d5 87 8d d3 87 a9 00 8d  27
85d8 : cf 87 8d de 87 ad ce 87  da
85e0 : f0 10 ad cb 87 f0 08 a9  31
85e8 : 8b 20 d2 ff 4c f2 85 20  eb
85f0 : cf ff ad cd 87 f0 03 4c  88
85f8 : aa 86 ad cb 87 f0 25 ad  ba
8600 : cf 87 c9 08 90 03 20 94  d1
8608 : 86 20 a0 81 90 10 a9 07  35
8610 : 38 ed cf 87 aa ad de 87  c6
8618 : 1d b9 87 8d de 87 ee cf  2b
8620 : 87 4c 56 86 ad cf 87 d0  4d
8628 : 0d 20 cf ff ae cc 87 f0  8a
8630 : 02 49 ff 8d de 87 0e de  a8
8638 : 87 90 06 20 a6 81 4c 44  bd
8640 : 86 20 a3 81 ee cf 87 ad  d6
8648 : cf 87 c9 08 90 08 a9 00  3e
8650 : 8d de 87 8d cf 87 ad d3  77
8658 : 87 cd dc 87 d0 23 ad d2  70
8660 : 87 cd db 87 d0 1b ae cd  f2
8668 : 87 d0 08 ae cb 87 f0 03  f2
8670 : 20 94 86 ad d9 87 cd dd  fe
8678 : 87 f0 11 ee d9 87 4c c9  38
8680 : 85 ee d2 87 d0 03 ee d3  aa
8688 : 87 4c f2 85 20 cc ff a9  5e
8690 : 02 4c c3 ff ad de 87 ae  f6
8698 : cc 87 f0 02 49 ff 20 d2  5f
86a0 : ff a9 00 8d cf 87 8d de  53
86a8 : 87 60 ae cc 87 f0 06 20  fd
86b0 : 9d 81 4c 56 86 20 a3 81  e7
86b8 : 4c 56 86 20 70 83 8e cc  cc
86c0 : 87 20 70 83 8e d2 87 20  c2
86c8 : 70 83 8e d9 87 a2 02 20  ae
86d0 : c6 ff 20 cf ff 8d db 87  83
86d8 : 20 cf ff 8d dd 87 a9 00  52
86e0 : 8d cb 87 4c 1d 87 20 70  2e
86e8 : 83 8e cc 87 20 70 83 8e  87
86f0 : d2 87 20 70 83 8e d9 87  bf
86f8 : 20 70 83 8e db 87 20 70  5e
8700 : 83 8e dd 87 a2 02 20 c9  81
8708 : ff a9 01 8d cb 87 ad db  35
8710 : 87 20 d2 ff ad dd 87 20  84
8718 : d2 ff 4c 1d 87 a9 00 8d  81
8720 : d8 87 a2 00 8e d5 87 8e  37
8728 : d0 87 ad d8 87 18 6d d9  e5
8730 : 87 0a aa bd e4 87 85 fb  b7
8738 : bd e5 87 85 fc ad d2 87  12
8740 : 0a 0a 0a 2e d5 87 18 65  5c
8748 : fb 85 fb ad d5 87 65 fc  e3
8750 : 85 fc a0 00 ae cb 87 f0  c5
8758 : 18 a2 35 78 86 01 b1 fb  4d
8760 : a2 37 86 01 ae cc 87 f0  b1
8768 : 02 49 ff 20 d2 ff 4c 7d  6c
8770 : 87 20 cf ff ae cc 87 f0  4c
8778 : 02 49 ff 91 fb c8 c0 08  6a
8780 : 90 d2 a5 fb 18 69 08 85  5a
8788 : fb 90 02 e6 fc ee d0 87  c2
8790 : ad d0 87 cd db 87 90 ba  f2
8798 : ee d8 87 ad d8 87 cd dd  47
87a0 : 87 b0 03 4c 22 87 20 cc  42
87a8 : ff a9 02 4c c3 ff 00 00  c2
87b0 : 00 fe fd fb f7 ef df bf  2c
87b8 : 7f 01 02 04 08 10 20 40  bb
87c0 : 80 11 12 13 14 15 16 17  20
87c8 : 11 00 00 00 00 00 00 00  da
87d0 : 00 00 00 00 00 00 00 00  d1
87d8 : 00 00 00 00 00 00 00 00  d9
87e0 : 00 00 00 00 e0 40 e1  ac
87e8 : 80 e2 c0 e3 00 e5 40 e6  84
87f0 : 80 e7 c0 e8 00 ea 40 eb  e1
87f8 : 80 ec c0 ed 00 ef 40 f0  3e
8800 : 80 f1 c0 f2 00 f4 40 f5  9c
8808 : 80 f6 c0 f7 00 f9 40 fa  f9
8810 : 80 fb c0 fc 00 fe b4 00  28
8818 : 34 00 58 77 50 00 ff ff  56
```

Listing 3. (Schluß)

```
Name : pm-sprite          c440 c540
-----------------------------------
c440 : ff ff ff 80 00 00 80 00  51
c448 : 00 80 00 00 80 00 00 80  92
c450 : 00 00 80 00 00 80 00 00  75
c458 : 80 00 00 80 00 00 80 00  eb
c460 : 00 80 00 00 80 00 00 80  aa
c468 : 00 00 80 00 00 80 00 00  8d
c470 : 80 00 00 80 00 00 80 00  03
c478 : 00 80 00 00 80 00 00 00  c1
c480 : ff ff ff 00 00 01 00 00  87
c488 : 01 00 00 01 00 00 01 00  ae
c490 : 00 01 00 00 01 00 00 01  23
c498 : 00 00 01 00 00 01 00 00  e1
c4a0 : 01 00 00 01 00 00 01 00  c6
c4a8 : 00 01 00 00 01 00 00 01  3b
c4b0 : 00 00 01 00 00 01 00 00  f9
c4b8 : 01 00 00 01 00 00 01 00  de
c4c0 : 80 00 00 80 00 00 80 00  53
c4c8 : 00 80 00 00 80 00 00 80  12
c4d0 : 00 00 80 00 00 80 00 00  f5
c4d8 : 80 00 00 80 00 00 80 00  6b
c4e0 : 00 80 00 00 80 00 00 80  2a
c4e8 : 00 00 80 00 00 80 00 00  0d
c4f0 : 80 00 00 80 00 00 80 00  83
c4f8 : 00 80 00 00 ff ff ff 00  38
c500 : 00 00 01 00 00 01 00 00  49
c508 : 01 00 00 01 00 00 01 00  2e
c510 : 00 01 00 00 01 00 00 01  a3
c518 : 00 00 01 00 00 01 00 00  61
c520 : 01 00 00 01 00 00 01 00  46
c528 : 00 01 00 00 01 00 00 01  bb
c530 : 00 00 01 00 00 01 00 00  79
c538 : 01 00 00 01 ff ff ff 00  5a
```

Listing 4.
Sprites für das Fenster



3D GRAFIK IN ECHTZEIT



Endlich können Sie dreidimensionale Körper komfortabel eingeben und ohne jeglichen Rechenaufwand auf dem Bildschirm darstellen. Jeder plastische Körper läßt sich in Echtzeit manuell oder automatisch in jede beliebige Richtung und jede gewünschte Achse drehen.

Eine der gefragtesten Anwendungen mit dem Computer ist wohl das Arbeiten mit der hochauflösenden Grafik. Sei es nun, um sich künstlerisch mit farbigen Malprogrammen (Koala-Painter oder Paint Magic) zu betätigen oder sich Konstruktionszeichnungen mit geeigneten Zeichenprogrammen (Hi-Eddi, Star-Painter oder Profi-Painter) zu erstellen. Doch auch die räumliche Darstellung dreidimensionaler Körper ist mit der geeigneten Software (z.B. Giga-CAD) kein Problem.

Doch damit lassen sich 3D-Grafiken leider nur aus einem bestimmten Blickwinkel betrachten. Um einen Körper um eine Achse zu drehen, ist ein enormer Speicherplatz- und Rechenaufwand nötig, der sehr zeitaufwendig werden kann und den auch nicht jedes Programm beherrscht. Hier nun greift »3D-Grafik-Master« ein.

Das Programm ermöglicht es, Körperdrehungen um beliebige Achsen fast in Echtzeit zu berechnen und anzuzeigen. Die Darstellung wird nur durch die Menge der im Körper enthaltenen Eckpunkte und Begrenzungslinien gebremst. Je weniger Linien zu zeichnen sind, desto schneller wird die Drehung. Bei bis zu etwa 40 Linien ist eine Rotation in Echtzeit möglich. Damit lassen sich bereits komplexe Körper darstellen (der Körper in Bild 1 ist beispielsweise aus 43 Punkten und 80 Linien aufgebaut). Mit steigender Zahl der Linien läßt sich bei der Rotation eine Zeitverzögerung nicht vermeiden.

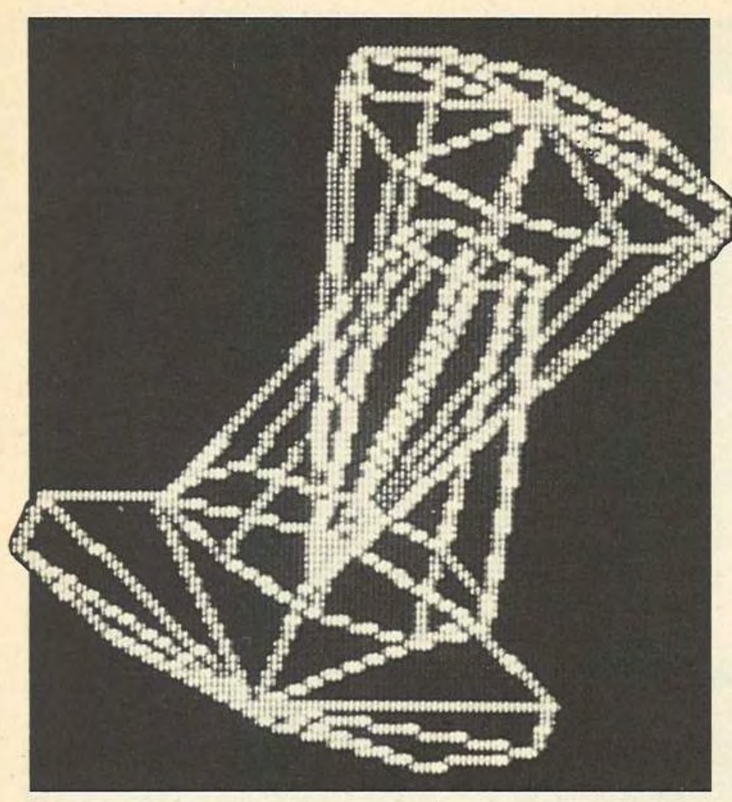


Bild 1. Drahtgittermodell eines dreidimensionalen Körpers

Die Menüpunkte

Nach dem Start des Programms mit
`LOAD"3D-GRAFIK-MASTER",8: RUN`
stehen nach dem Laden der einzelnen Programmteile und einer Initialisierungszeit von etwa 50 Sekunden (Aufbau einer Tabelle) folgende Menüpunkte zur Auswahl, die sich entweder mit den Cursortasten oder durch direkte Anwahl mit den Zifferntasten <1> bis <8> ansprechen lassen:

<1> Eingeben als 3D-Zeichnung

Dieser Punkt dient zum Eingeben eines dreidimensionalen Körpers. Sollte vorher bereits ein Körper im Speicher existieren, wird dieser zuerst vom Programm gezeichnet. Damit ist es möglich, neue Verbindungen an bereits existierenden Körpern anzubringen.

Mit den Tasten <X>, <Y>, <Z>, <SHIFT+X>, <SHIFT+Y>, <SHIFT+Z> und den beiden Cursor-Tasten kann nun eine Linie frei im Raum bewegt werden. <RETURN> übernimmt die gezeichnete Linie in den Speicher. <SHIFT+CLR/HOME> löscht das Bild, <CLR/HOME> positioniert den Grafik-Cursor auf den Koordinatennullpunkt. Durch Drücken der <*>-Taste kann der sich bewegende Endpunkt der Linie gewechselt werden. Auf Wunsch blendet <K> ein Koordinatenkreuz in die Grafik ein oder aus. <L> setzt den Grafik-Cursor auf den zuletzt mit <RETURN> übernommenen Punkt. <P> vereinigt Anfangs- und Endpunkt der Linie miteinander in der aktuellen Position des Grafik-Cursors (ab hier startet eine neue Linie). Mit der Taste <A> läßt sich die X-, Y- und Z-Koordinate des Grafik-Cursors abfragen. Der Grafik-Cursor bewegt sich in Schritten, die jeweils einen Abstand von fünf Bildpunkten haben. Eine Änderung ist mit den Tasten <0> bis <9> möglich, wobei <0> einer Schrittweite von 10 Pixel (Bildschirmpunkten) entspricht.

Mit den Tasten <+> und <—> wird der Sichtwinkel, unter dem der Körper erscheint, verändert.

Ist der Körper fertig gezeichnet, erfolgt die Rückkehr zum Hauptmenü mit den Tasten <←> oder <M>.

<2> Eingeben als Koordinatentripel

Hier kann die Eingabe der Koordinaten eines Körpers in numerischer Form erfolgen (Bild 2). Die Koordinaten dürfen Werte von —40 bis +40 annehmen. Eine Überprüfung findet im Programm statt. <CLR/HOME> setzt den Grafik-Cursor in die erste Spalte der ersten Zeile, <SHIFT+CLR/HOME> löscht den Speicher. Mit den Cursortasten kann der Speicher durchgeblättert werden. Sollte sich der Cursor in der Zeile befinden, in der der letzte Punkt angezeigt wird, läßt sich der Punktespeicher mit <RETURN> um einen Punkt erweitern, und neue Koordinaten können eingegeben werden. Die maximale Anzahl einzugebender Punkte und Verbindungslinien beträgt jeweils 255.

<SHIFT+INST/DEL> schafft ebenfalls Platz für einen weiteren Punkt, <INST/DEL> löscht einen Punkt. Ist ein zu löschender Punkt mit einem anderen Punkt durch eine Linie verbunden, läßt sich dieser Punkt nicht löschen!

Mit <F7> gelangt man in den Linien-Eingabemodus. Hier gibt man die Punktenummern des Anfangs- wie auch

```
Koordinatentripel eingeben:

 Anzahl der Punkte: 43
 Anzahl der Linien: 80
-----------------------------------
Nr.  1   ==>  x=-40    y= 0
Nr.  2   ==>  x=-30    y= 39
Nr.  3   ==>  x=-20    y= 24
Nr.  4   ==>  x= 20    y= 9
Nr.  5   ==>  x= 40    y= 0
Nr.  6   ==>  x= 35    y= 29
Nr.  7   ==>  x=-25    y= 0
Nr.  8   ==>  x=-30    y= 32
Nr.  9   ==>  x=-20    y= 20
Nr. 10   ==>  x= 20    y= 8
Nr. 11   ==>  x= 35    y= 24
Nr. 12   ==>  x=-30    y= 12
Nr. 13   ==>  x=-20    y= 7
Nr. 14   ==>  x= 20    y= 3
Nr. 15   ==>  x= 35    y= 9
Nr. 16   ==>  x=-█0    y=-13
-----------------------------------
```

Bild 2. Auch die direkte Eingabe und Änderung von Raumkoordin durch direkte Zahleneingabe möglich

des Endpunktes ein. <SHIFT+INST/DEL> schafft Platz für eine neue Verbindungslinie, <INST/DEL> löscht eine Verbindungslinie.

Mit <F7> gelangt man wieder in den Punkte-Eingabemodus. Von hier aus kann auch wieder mit <←> oder <M> in das Hauptmenü zurückgesprungen werden.

<3> Eingeben von Rotationskörpern

Nach der Anwahl dieses Menüpunkts wird man gefragt, ob die Koordinaten eines möglicherweise im Speicher stehenden Körpers gelöscht werden sollen. Beantwortet man diese Frage mit <N>, zeichnet das Programm den nächsten erstellten Körper zu dem bestehenden dazu. Nach Klärung dieser Frage erfolgt das Zeichnen eines 80 x 40 Punkte großen Eingabefeldes. Das Programm erstellt nun innerhalb dieses Eingabefeldes eine Hilfsmaske (Bild 3) mit einem Punktabstand von fünf Pixel.

Innerhalb dieses Eingabefeldes kann nun eine Konturenlinie des Körpers erstellt werden, die um eine Drehachse (hier das Maßband) gedreht wird (Bild 3). Wie oft dies zu ge-

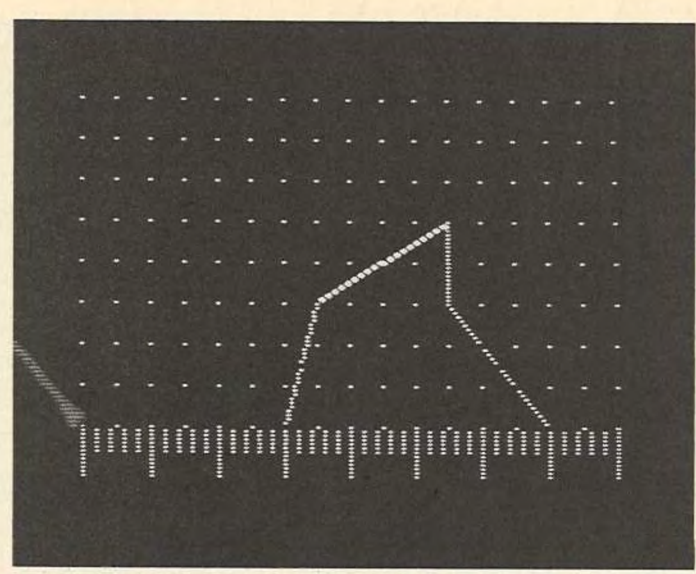
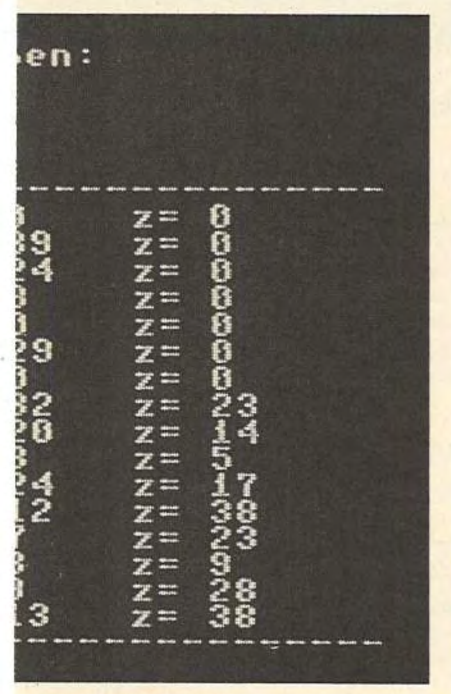


Bild 3. Die Hilfsmaske beim Erstellen von Rotationskörpern mit angedeutetem Körper



aten ist bequem

schehen hat, ist nach Abschluß des Konturenlinienzeichnens anzugeben. Das Programm verbindet anschließend die entsprechenden Knickpunkte der gedrehten Konturenlinie miteinander, so daß ein geschlossener Körper entsteht (Bild 4).

Das Eingeben der Konturenlinie geschieht mit den Tasten <X>, <Y>, <CRSR>-hoch/unten, <CRSR>-rechts/links, <RETURN>, <*>, <SHIFT+CLR/HOME>, <CLR/HOME>, <L> und <P>. Mit <X> und <Y> sowie den Cursortasten kann man den Grafik-Cursor bewegen. Die Übernahme einer Linie erfolgt mit <RETURN>. Durch Druck auf <*> kann der sich bewegende Punkt der Linie gewechselt werden. <SHIFT+CLR/HOME> löscht das Eingabefeld und den Punktespeicher und baut das Eingabefeld wieder neu auf. <CLR/HOME> setzt den Grafik-Cursor auf den geometrischen Nullpunkt des Körpers, <L> setzt ihn auf den zuletzt übernommenen Punkt und <P> vereinigt Anfangs- und Endpunkt der Linie in der momentanen Position des Grafik-Cursors. Die Schrittweite läßt sich mit den Zahlentasten <0> bis <9> pixelweise verändern, wobei <0> einer Schrittweite von 10 Pixel entspricht. Ist die Konturenlinie fertiggestellt, beendet <SPACE> die Eingabe.

Nun fragt das Programm, bis zu welchem Winkel der Körper berechnet werden soll. Hierbei entsprechen 360 Grad einem kompletten Körper, 180 Grad wären genau ein halber Körper.

Das Programm wartet jetzt darauf, daß die Anzahl der Facetten eingegeben wird. Hier gilt: Je höher die Facettenzahl, desto runder erscheint der Körper, desto langsamer erfolgt aber dann auch die Drehung.

Soll z.B. ein halbierter Körper entstehen, der sonst aber »rund« sein soll, so muß ein Winkel von 180 Grad und eine Facettenzahl von etwa 20 angegeben werden.

Ist alles korrekt, ist die Frage »Alles ok?« mit <RETURN> zu beantworten, ansonsten geben Sie bitte <N> ein.

Der Computer berechnet nun die einzelnen Teilstücke und Verbindungslinien der Figur. Danach kehrt das Programm automatisch in das Hauptmenü zurück.

<4>/<5> Koordinaten laden und speichern

In diesen Menüpunkten darf ein höchstens 13 Zeichen langer Name angegeben werden, an den das Programm noch ».3D« anhängt (dies dient der Wiedererkennung der einzelnen Dateien auf Diskette).

<6> Directory

Hier lassen sich alle sequentiellen Disketten-Files anzeigen, die als Datendateien (Name.3D) gekennzeichnet sind.

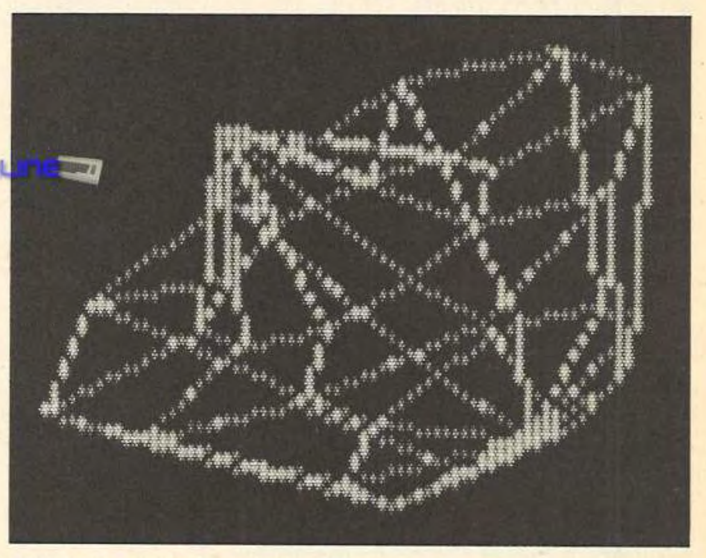


Bild 4. So sieht der mit der Funktion »Rotation« erstellte Körper plastisch aus

Durch Druck auf eine beliebige Taste wird die Ausgabe angehalten. Nach der Anzeige aller berechtigten Dateien zeigt das Programm den Fehlerkanal des Disketten-Laufwerks an. Im Normalfall also »00,OK,00,00«. Ein weiterer Tastendruck führt zurück ins Hauptmenü.

<7> Drehen

Dies ist einer der letzten und wichtigsten Menüpunkte. Er erlaubt das Drehen einer 3D-Figur fast in Echtzeit (bei nicht zu vielen Linien; je mehr Linien, desto langsamer wird die Drehung).

Beim Start befindet man sich im »Demo-Modus«. Hier wird ein Körper um alle drei Achsen (X-, Y- und Z-Achse) gleichzeitig gedreht. Abschalten kann man die selbständige Drehung mit <E>. Mittels <X>, <Y> und <Z> bzw. <SHIFT+X>, <SHIFT+Y> und <SHIFT+Z> läßt sich ein Körper von Hand in jede Lage drehen. <+> und <-> verändern dabei den Sichtwinkel (von 0 bis 92 Grad). Der Demo-Modus läßt sich durch Druck auf <D> erneut starten. Die Tastaturabfrage ist so gestaltet, daß auch alle

Kurzinfo: 3D-Grafik

Programmart: Grafikprogramm zur Darstellung von dreidimensionalen Körpern
Laden: LOAD "3D-GRAFIK-MASTER",8
Start: Mit RUN
Besonderheiten: Körperdrehungen um beliebige Achsen. Anzeige fast in Echtzeit. Hilfsmaske bei Erstellen von Rotationskörpern. Floppy-Schnell-Lader müssen abgeschaltet sein.
Programmautor: Jesko Schwarzer

Tasten gleichzeitig gedrückt werden dürfen. <RETURN> führt wieder zurück ins Hauptmenü.

<8> Exit

Zu diesem Punkt ist nicht viel zu sagen. Hiermit wird der Editor verlassen. Ein Warmstart ist mit der Eingabe von GOTO 60030 möglich.

Eingabehinweise

Bitte geben Sie zuerst Listing 1 ein und speichern es unter dem Namen »3D-Grafik-Master«. Dies ist das spätere Ladeprogramm, das die anderen Teile automatisch nach dem Start mit RUN nachlädt.

Doch vor dem Start tippen Sie bitte noch die Listings 2 und 3 (beide mit dem MSE) sowie Listing 4 und 5 ein und speichern diese.

Listing 3 (»3D.SPRT«) enthält die Form des Cursorsprites. Sie kann beliebig verändert werden. Es ist nur darauf zu achten, daß die Figur auf die rechte untere Ecke zeigt und die Anfangsadresse $5BC0 (26368) lautet. Das Listing 4 muß unter dem Namen »3D.EDIT« gespeichert werden. Beachten Sie unbedingt, daß in diesen Listings die Zeilen »20330« und »21630« nur mit Basic-Abkürzungen (siehe Handbuch) einzugeben sind. Im anderen Fall würden die Zeilen zu lang und nicht vom C64 akzeptiert werden.

Das Maschinenprogramm (Listing 2) läuft ohne weiteres auch ohne den Editor (Listing 4). Wie ein eigenes Programm aussieht, das mit Unterstützung von Listing 2 bewegte Grafiken darstellt, können Sie mit Hilfe des Demo-Programms (Listing 5) sehen. Fertige Konstruktionen (z.B. Bild 6 u. 7) finden Sie auf der Service-Diskette zum Heft.

Das Programm

Der Hauptteil des Programms ist sicher die Drehroutine. Sie dreht einen Körper und stellt ihn in Parallelprojektion dar. Dabei wird mit zwei Grafikbildschirmen gearbeitet. Der erste liegt bei $6000 (Farbspeicher bei $5C00) und der zweite bei $E000 (Farbspeicher im RAM unter dem I/O-Bereich ab $DC00).

Die LINE-Routine ist eine der schnellsten uns bekannten. Sie setzt nahezu 13000 Punkte pro Sekunde. Sollte der Sonderfall einer Horizontal- oder Vertikal-Linie eintreten, verzweigt das Programm zu noch schnelleren Unterroutinen. Nach dem Umschalten zwischen den Grafikbildschirmen wird der andere, nicht sichtbare Grafikbildschirm gelöscht (nur der Ausschnitt vom 64. Pixel bis zum 255. Pixel, da ausschließlich in diesem Bereich gezeichnet wird).

Koordinaten und Verbindungslinien

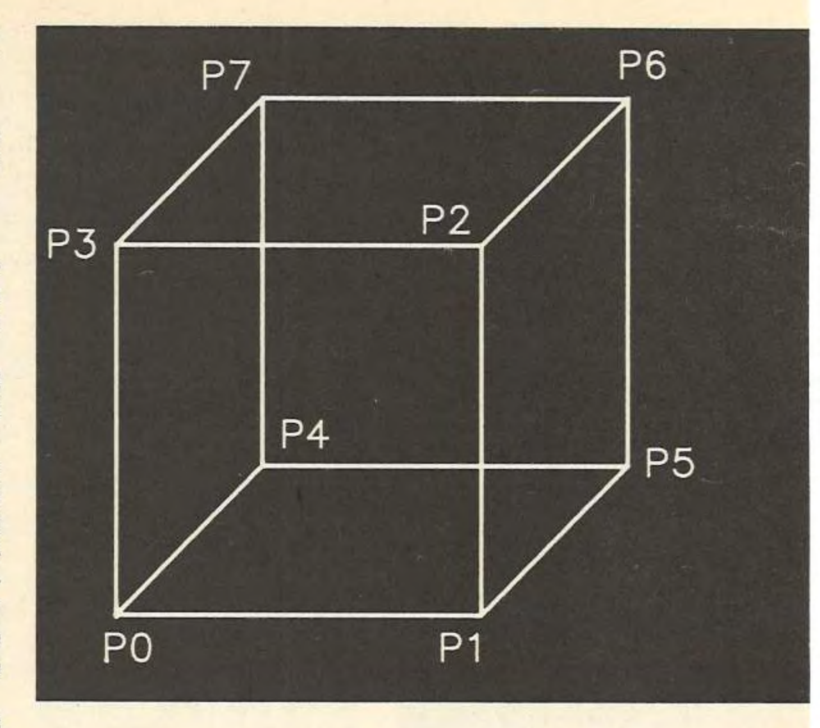
Jeder Punkt wird im Format X/Y/Z im RAM gespeichert. Hierbei liegen die X-Koordinaten ab $4C00 (19456), die Y-Werte ab $4D00 (19712) und die der Z-Punkte ab $4E00 (19968). Es können maximal 255 Punkte abgelegt werden. Die Anzahl steht in Speicherstelle 3. Ein Beispiel: Die Koordinaten eines Würfels (Bild 5):

Zu negativen Zahlen muß der Wert 256 addiert werden!

19456	X	19712	Y	19968	Z
+0	226	+0	226	+0	226
+1	30	+1	226	+1	226
+2	30	+2	226	+2	30
+3	226	+3	226	+3	30
+4	226	+4	30	+4	226
+5	30	+5	30	+5	226
+6	30	+6	30	+6	30
+7	226	+7	30	+7	30

Die Zahl der Punkte beträgt 8, also POKE 3,8

Die Verbindungsvorschrift liegt ab $5100 (20736) für den Startpunkt (PS) und ab $5200 (20992) für den Endpunkt (PE). Es wird dann beim späteren Drehen eine Linie von PS nach PE gezeichnet, wenn sich die entsprechenden Punktnummern in den Speicherzellen befinden. Z.B. der Würfel aus Bild 5 von oben:



Nr.	PS	PE	
0	0	1	Da 12 Linien gezeichnet wer-
1	1	2	den sollen (0 bis 11), muß
2	2	3	Speicherstelle 4 eine 12
3	3	0	enthalten.
4	4	5	(POKE 4,12)
5	5	6	
6	6	7	
7	7	4	
8	0	4	
9	1	5	
12	2	6	
11	3	7	

Im RAM sieht das dann so aus:

20736 (PS)	Inhalt	20992 (PE)	Inhalt
+0	0	+0	1
+1	1	+1	2
+2	2	+2	3
+3	3	+3	0
+4	4	+4	5
+5	5	+5	6
+6	6	+6	7
+7	7	+7	4
+8	0	+8	4
+9	1	+9	5
+10	2	+10	6
+11	3	+11	7

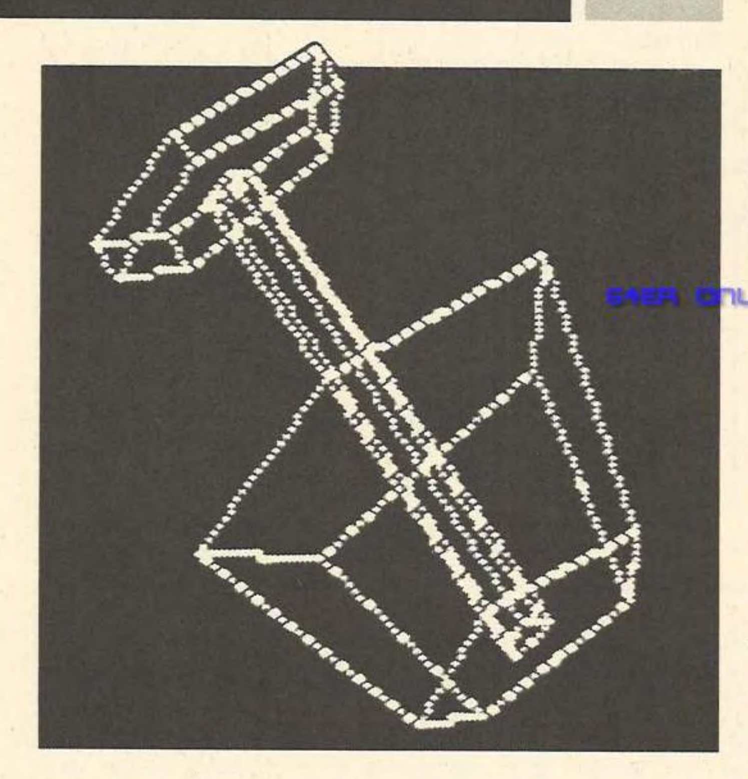
Maschinenprogramm (Funktion)

Damit doppelt eingegebene Punkte nicht doppelt berechnet werden müssen, existiert eine Maschinenroutine, die die Koordinaten selbständig im Speicher ablegt und die Punktenummer an den Benutzer zurückgibt. Die Routine, die diese Aufgabe übernimmt, heißt PSEARCH und liegt ab $400C. Sie hat das Format »SYS 2↑14+12,X,Y,Z«.

X, Y und Z sind die Koordinaten eines jeden Punktes. Die Punktenummer wird in Speicherstelle 782 übergeben. Eine praktische Anwendung findet sie im Basic-Listing ab Zeile

```
P0(-30,-30,-30)
P1( 30,-30,-30)
P2( 30,-30, 30)
P3(-30,-30, 30)
P4(-30, 30,-30)
P5( 30, 30,-30)
P6( 30, 30, 30)
P7(-30, 30, 30)
```

Bild 5. Ein Würfel im Koordinatensystem



30630. Die Adresse der PSEARCH-Routine ist in der Variablen SE gespeichert. Hier, im Rotationskörper-Berechnungsprogramm, kann es vorkommen, daß zwei Punkte gleiche Koordinaten haben. Um zu verhindern, daß unnötigerweise beide berechnet und gezeichnet werden, existiert diese Routine.

Ein ähnliches Programm existiert auch für Verbindungslinien. Damit keine Linie doppelt gezeichnet wird, können die Punktenummern der LSEARCH-Routine ab $400F (im Basic-Programm die Routine mit der Variablen VP) übergeben werden. Syntax: »SYS 2↑14+15,PS,PE«. Speicherzelle 4 (Anzahl der Linien) wird in jedem Fall angepaßt.

Sollte in eine bestehende Punktetabelle ein weiterer Punkt mit einer ganz bestimmten Punktnummer eingefügt werden, dann benutzt man die INSTP-Routine ab $401E. Sie schafft Platz für weitere Punkte durch: »SYS 2↑14+30,PN«. PN stellt die Nummer des einzusetzenden Punktes dar. Die Tabelle der Verbindungslinien wird dabei angepaßt. Das ist wichtig, da der Körper sonst nicht mehr korrekt gezeichnet werden kann.

Ab $4021 existiert noch die PDEL-Funktion. Sie entfernt einen Punkt aus dem Speicher, wenn dieser Punkt nicht mit einem anderen durch eine Linie verbunden ist. Das bedeutet im Klartext: Existiert in der PS- oder PE-Tabelle diese Punktnummer, wird der Punkt nicht gelöscht. Das Format lautet: »SYS 2↑14+33,PN«. Speicherzelle 3 paßt sich entsprechend an. Praktische Anwendung finden beide Routinen (PINST und PDEL) ab Zeile 20320 in Listing 4.

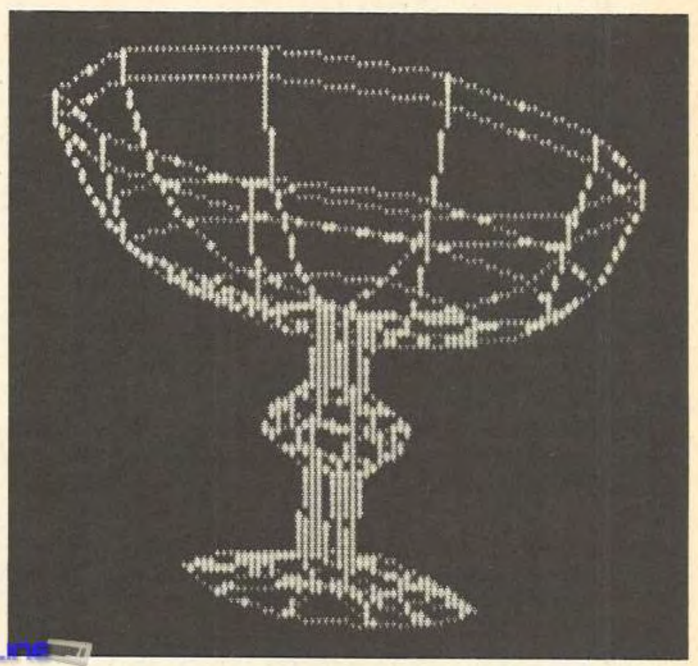


Bild 6 und 7. Diese beiden Konstruktionen finden Sie auf der Service-Diskette zum Heft

Verbindungslinien können mit der VINST-Routine ($4024) eingesetzt werden. Das Format: »SYS 2↑14+36,LN«. Speicherzelle 4 (Anzahl der Linien) paßt sich automatisch an.

Die VDEL-Routine ($4027) löscht Verbindungsroutinen. Format: »SYS 2↑14+39,LN«. Speicherstelle 4 wird automatisch erniedrigt. Das Basic-Listing benutzt beide Routinen ab Zeile 21620.

Sind alle Koordinaten im Speicher abgelegt und die Speicherstellen 3 und 4 richtig gesetzt, darf die Initialisierungs- ($4000) und dann die Drehroutine ($042A) aufgerufen werden. Die INIT-Routine überprüft, ob alle internen Tabellen aufgestellt wurden. Wenn nicht (erster Aufruf oder Speicherstelle 10023 < >0), holt sie das nach. Sie schaltet die $E000-Grafik ein, löscht diese und setzt den Farbspeicher. Danach wird die LINE-Routine auf die richtige Grafikseite (page) bei $6000 gelenkt und ins Basic zurückgesprungen.

Die Drehroutine fragt die Tastatur ab, berechnet die Drehung, die 2D-Koordinaten für den Bildschirm und zeichnet die Figur, bis <RETURN> gedrückt wird. Z.B.: »SYS 2↑14: SYS 2↑14+42« – Siehe auch Zeile 60020 in Listing 4.

Sollte jemand Lust verspüren, den Körper zu speichern, muß erst die Speicherstelle 3 und dann 4 ausgelesen werden. Nun folgt das Senden der X-, Y- und Z-Koordinaten des ersten Punktes, danach die des zweiten usw. Anschließend kommt die Verbindungsvorschrift. PS/PE der ersten Linie usw., bis die in Speicherstelle 4 enthaltene Anzahl der Linien erreicht ist. Für die Besitzer eines Disketten-Laufwerks übernimmt die Routine ab $4018 das Speichern: »SYS 2↑14+24,1,8,1,"NAME,S":CLOSE1«.

Für das Laden ist das Maschinenprogramm ab $401B zuständig: »SYS 2↑14+27,1,8,0,"NAME,S":CLOSE1«.

Besitzer eines (möglicherweise nicht kompatiblen) Schnelladers können dafür den Programmteil ab Zeile 40000 in Listing 4 verwenden.

Da im Editor auch Linien gezogen und Punkte gesetzt werden, müssen die Einsprünge für diese Routinen auch vorhanden sein:

$4012 – PLOT: »SYS 2↑14+18,X,Y« (2D-Koordinaten)
$4003 – LINE: »SYS 2↑14+3,X1,X2,Y1,Y2«
$4006 – INV: »SYS 2↑14+6« (Ein)
$4015 – NRM: »SYS 2↑14+21« (Aus)

Schulung des Vorstellungsvermögen

Die Grafikbefehle beziehen sich auf die Grafik ab $6000, wenn vor der Benutzung »SYS 2↑14:POKE 56576,2« eingegeben wurde.

Eine OLD-Funktion ist auch noch vorhanden. Sie wird mit »SYS 2↑14+9« aufgerufen und rettet ein mit Reset oder NEW gelöschtes Programm.

Ein Tip noch am Rande:

Sollen Linien zwar gezeichnet, aber nicht gedreht werden, müssen diese Punkte (deren Koordinaten) in der Punktetabelle »hinten« liegen. Die Speicherstelle 3 gibt weiterhin die Anzahl der Punkte an, die gedreht werden sollen. Punkte, die weiter »hinten« in der Tabelle liegen, können zwar mit Linien verbunden werden, sind aber beim Drehen nicht relevant.

Der 3D-Grafik-Master (der sich übrigens auch in eigene Programme einbinden läßt) ist sehr gut geeignet, um das räumliche Erkennungsvermögen und die Vorstellungskraft zu schulen und zu unterstützen. Ein Beispiel dafür wäre etwa der Einsatz in der Schule, wenn es darum geht, im Zeichenunterricht plastische Körper den Schülern verständlich darzustellen. Doch auch für das eigene Vorstellungsvermögen ist es eine wertvolle Stütze.

Das Programm benutzt zur Darstellung das Drahtgittermodell (Bild 1). Aus Geschwindigkeitsgründen wurde auf eine Möglichkeit zum Verbergen verdeckter Kanten (Hidden-Line) verzichtet, da die Routinen sonst in ihrer Verarbeitungsgeschwindigkeit sehr stark verlangsamt würden.

Nach der Eingabe der Körper, die über direktes Zeichnen im Raum bis zur Eingabe über Koordination reicht (Bild 2), baut das Programm innerhalb kürzester Zeit die nötigen Verbindungen im Speicher auf. Gleich danach kann der fertige Körper dreidimensional dargestellt werden. Es ist außerdem jederzeit möglich, bestehende Verbindungen und Punkte zu löschen oder neue hinzuzufügen. Der zeitliche Aufwand dafür belastet den Anwender nur sehr gering, so daß ein effektives Arbeiten mit dem 3D-Grafik-Master möglich ist.

Einmal damit gearbeitet, werden Sie nicht so schnell die Freude an diesem gelungenen Programm verlieren.

(Jesko Schwarzer/ag)

```
100 REM"********************************    <086>
110 REM"********************************    <096>
120 REM"**{27SPACE}**                       <077>
130 REM"**{8SPACE}3D - MASTER{8SPACE}**     <204>
140 REM"**{8SPACE}TTTTTTTTTTT{8SPACE}**     <022>
150 REM"**{2SPACE}VON : JESKO SCHWARZER{4S
    PACE}**                                 <217>
160 REM"**{27SPACE}**                       <117>
280 REM"********************************    <012>
290 REM"********************************    <022>
300 :                                       <022>
330 A=A+1:ON A GOTO 1000,2000,3000          <244>
1000 PRINT"1.MASCHINENPROGRAMM              <106>
1010 LOAD"3D.CODE",8,1                      <167>
1020 :                                      <234>
2000 PRINT"2.SPRITE                         <152>
2010 LOAD"3D.SPRT",8,1                      <248>
2020 :                                      <218>
3000 PRINT"3.EDITOR                         <112>
3010 LOAD"3D.EDIT",8                        <032>
3020 :                                      <202>
```

Listing 1. Das Ladeprogramm »3D-Grafik-Master«. Beachten Sie bitte die Eingabehinweise auf Seite 159.

```
Name : 3d.code              4000 4bc8
-------------------------------------
4000 : 4c 50 43 4c fc 4a 4c de   e0
4008 : 40 4c b2 4b 4c 3e 40 4c   d5
4010 : 92 40 4c e4 4a 4c db 40   69
4018 : 4c ed 40 4c 39 41 4c 85   ce
4020 : 41 4c cc 41 4c 25 42 4c   72
4028 : 4b 42 20 db 40 78 a9 30   e7
4030 : 85 01 20 01 44 20 a4 45   c0
4038 : 20 82 42 4c 2e 40 20 fd   15
4040 : ae 20 9e b7 86 05 20 fd   aa
4048 : ae 20 9e b7 86 06 20 fd   ba
4050 : ae 20 9e b7 86 0a 8a a4   d9
4058 : 03 f0 0c a0 00 d9 00 4e   56
4060 : f0 1d c8 c4 03 90 f6 99   6d
4068 : 00 4e a5 05 99 00 4c a5   af
4070 : 06 99 00 4d e6 03 a9 80   1b
4078 : 85 05 85 06 85 0a 60 a5   17
4080 : 05 d9 00 4c d0 07 a5 06   e3
4088 : d9 00 4d f0 e9 a5 0a 4c   5f
4090 : 62 40 20 fd ae 20 9e b7   b0
4098 : 86 05 20 fd ae 20 9e b7   3e
40a0 : 8a a4 04 f0 11 a0 00 d9   65
40a8 : 00 51 f0 19 d9 00 52 f0   79
40b0 : 1f c8 c4 04 90 f1 99 00   e4
40b8 : 51 a5 05 99 00 52 e6 04   87
40c0 : a9 80 85 05 60 a5 05 d9   a6
40c8 : 00 52 f0 f4 8a 4c b1 40   1e
40d0 : a5 05 d9 00 51 f0 e9 8a   c8
40d8 : 4c b1 40 a9 05 2c a9 45   25
40e0 : 8d 41 49 8d 47 49 8d f4   f1
40e8 : 48 8d fd 48 60 20 fd ae   dc
40f0 : 20 be e1 a6 b8 86 13 20   09
40f8 : 13 e1 a5 03 20 d2 ff a5   a9
4100 : 04 20 d2 ff a0 00 b9 00   ba
4108 : 4c 20 d2 ff b9 00 4d 20   2a
4110 : d2 ff b9 00 4e 20 d2 ff   81
4118 : c8 c4 03 90 e9 a0 00 b9   2c
4120 : 00 51 20 d2 ff b9 00 52   9e
4128 : 20 d2 ff c8 c4 04 90 ef   59
4130 : a9 00 85 13 85 99 4c cc   8d
4138 : ff 20 fd ae 20 be e1 a6   69
4140 : b8 86 13 20 1e e1 20 24   be
4148 : e1 85 03 20 24 e1 85 04   20
4150 : a0 00 20 13 ee 99 00 4c   af
4158 : 20 13 ee 99 00 4d 20 13   02
4160 : ee 99 00 4e c8 c4 03 90   c5
4168 : e9 a0 00 20 13 ee 99 00   b5
4170 : 51 20 13 ee 99 00 52 c8   e8
4178 : c4 04 90 ef a9 00 85 13   37
4180 : 85 99 4c cc ff 20 fd ae   d5
4188 : 20 9e b7 e0 ff d0 01 60   4c
4190 : 86 3f a6 03 a4 03 ca bd   c9
4198 : 00 4c 99 00 4c bd 00 4d   72
41a0 : 99 00 4d bd 00 4e 99 00   1d
41a8 : 4e 88 c4 3f d0 e8 e6 03   49
41b0 : a6 04 8a f0 da ca bd 00   14
41b8 : 51 c5 3f 90 03 fe 00 51   99
41c0 : bd 00 52 c5 3f 90 eb fe   f1
41c8 : 00 52 b0 e6 20 fd ae 20   e7
41d0 : 9e b7 e4 03 90 01 60 8a   8b
41d8 : a6 04 f0 fa ca dd 00 51   5a
41e0 : f0 f4 dd 00 52 f0 ef e0   f0
41e8 : 00 d0 f1 85 3f a8 aa e8   33
41f0 : bd 00 4c 99 00 4c bd 00   4d
41f8 : 4d 99 00 4d bd 00 4e 99   04
4200 : 00 4e c8 c4 03 d0 e8 c6   da
4208 : 03 a2 00 bd 00 51 c5 3f   34
```

Listing 2. »3D-CODE« – Maschinenroutinen zur schnellen Grafik-Animation (bitte mit dem MSE eingeben)

```
4210 : 90 03 de 00 51 bd 00 52  81
4218 : c5 3f 90 03 de 00 52 e8  0a
4220 : e4 04 90 e7 60 20 fd ae  84
4228 : 20 9e b7 e4 04 90 01 60  ab
4230 : 86 3f a6 04 a4 04 88 b9  80
4238 : 00 51 9d 00 51 b9 00 52  d0
4240 : 9d 00 52 ca c4 3f d0 ee  33
4248 : e6 04 60 20 fd ae 20 9e  5f
4250 : b7 e4 04 90 01 60 8a a8  1b
4258 : c8 b9 00 51 9d 00 51 b9  ba
4260 : 00 52 9d 00 52 e8 c4 04  78
4268 : d0 ee c6 04 60 58 68 68  1d
4270 : 68 68 a9 15 8d 18 d0 a9  4a
4278 : 1b 8d 11 d0 a9 03 8d 00  a1
4280 : dd 60 a9 37 85 01 a9 00  e6
4288 : 85 8d a9 00 8d 00 dc ae  e8
4290 : 01 dc e8 f0 27 a0 09 b9  67
4298 : 5f 4b 8d 00 dc ad 01 dc  f9
42a0 : 49 ff 39 55 4b f0 12 98  98
42a8 : 0a aa bd 69 4b 8d b7 42  28
42b0 : bd 6a 4b 8d b8 42 20 ff  45
42b8 : ff 88 10 db a5 9b f0 09  88
42c0 : 20 cf 42 20 e9 42 20 03  94
42c8 : 43 60 a9 ff 85 8d 60 a6  39
42d0 : 05 a5 8d d0 0a e8 e0 db  48
42d8 : 90 02 a2 80 86 05 60 ca  ca
42e0 : e0 80 b0 02 a2 da 86 05  92
42e8 : 60 a6 06 a5 8d d0 0a e8  2b
42f0 : e0 db 90 02 a2 80 86 06  77
42f8 : 60 ca e0 80 b0 02 a2 da  61
4300 : 86 06 60 a6 0a a5 8d d0  1c
4308 : 0a e8 e0 db 90 02 a2 80  de
4310 : 86 0a 60 ca e0 80 b0 02  e6
4318 : a2 da 86 0a 60 a9 ff 85  69
4320 : 9b 60 a9 00 85 9b 60 a6  5a
4328 : bf e8 e0 18 90 02 a2 17  68
4330 : 86 bf 4c 3e 43 a6 bf ca  6f
4338 : 10 02 a2 00 86 bf 38 a9  8d
4340 : 17 e5 bf 09 80 8d 7d 44  4e
4348 : a5 bf 09 80 8d 8e 44 60  3e
4350 : 78 a9 3b 8d 11 d0 a9 78  4d
4358 : 8d 18 d0 a9 00 8d 00 dd  83
4360 : a9 56 8d d6 4a 8d e5 48  ac
4368 : 20 de 40 a9 80 85 05 85  90
4370 : 06 85 0a 85 9b a9 30 85  3f
4378 : 01 a0 00 84 f7 84 f9 a9  39
4380 : 60 a2 e0 85 f8 86 fa a2  0f
4388 : 20 98 91 f7 91 f9 c8 d0  05
4390 : f9 e6 f8 e6 fa ca d0 f2  46
4398 : a0 fa a9 10 99 ff db 99  5e
43a0 : f9 dc 99 f3 dd 99 ed de  0c
43a8 : 99 ff 5b 99 f9 5c 99 f3  1c
43b0 : 5d 99 ed 5e 88 d0 e5 20  08
43b8 : 3e 43 a9 00 a0 60 a2 e0  5c
43c0 : 85 f7 84 f8 86 fa aa 18  9c
43c8 : a0 00 98 65 f7 9d 00 58  58
43d0 : 08 a5 f8 69 00 9d 00 56  b0
43d8 : 28 a5 fa 69 00 9d 00 57  5a
43e0 : e8 c8 c0 08 90 e4 a5 f7  14
43e8 : 69 3f 85 f7 08 a5 f8 69  b5
43f0 : 01 85 f8 28 a5 fa 69 01  d1
43f8 : 85 fa e0 c8 90 ca 4c ce  7a
4400 : 44 a2 00 86 f7 86 f9 a5  4d
4408 : 05 85 f8 c9 c3 90 02 e9  e4
4410 : 5b 69 17 85 fa bd 00 4d  cf
4418 : a8 bd 00 4e 84 8b 85 8c  3d
4420 : b1 f9 a4 8c 71 f7 85 fd  71
4428 : 38 b1 f9 a4 8b f1 f7 85  7f
4430 : 8c a5 06 85 f8 c9 c3 90  cf
4438 : 02 e9 5b 69 17 85 fa bd  38
4440 : 00 4c a8 84 8b b1 f9 a4  98
4448 : 8c 71 f7 85 c1 38 b1 f9  d4
4450 : a4 8b f1 f7 85 c2 a5 0a  4e
4458 : 85 f8 c9 c3 90 02 e9 5b  bc
4460 : 69 17 85 fa a4 c1 b1 f9  28
4468 : a4 fd 71 f7 85 fd 38 b1  f3
4470 : f9 a4 c1 f1 f7 18 a8 b9  c0
4478 : 00 db a8 b9 00 ff 65 fd  59
4480 : 18 69 a0 9d 00 4f 38 a9  d7
4488 : 64 e5 c2 38 f9 00 ff 9d  71
4490 : 00 50 e8 e4 03 b0 03 4c  e9
4498 : 07 44 a9 00 85 3f a6 3f  97
44a0 : bd 00 51 a8 b9 00 4f 85  ab
44a8 : ac b9 00 50 85 ae bd 00  00
44b0 : 52 a8 b9 00 4f 85 bb b9  48
44b8 : 00 50 85 bd a9 00 85 ad  05
44c0 : 85 bc 20 24 48 e6 3f a5  34
44c8 : 3f c5 04 90 d1 60 ad ff  d4
44d0 : 03 f0 05 a9 37 85 01 60  26
44d8 : a9 37 85 01 a9 0b 8d 11  ea
44e0 : d0 a9 03 8d 30 d0 a9 00  28
44e8 : 85 8d 85 8e 85 f7 85 3f  14
44f0 : a9 80 85 f8 a5 8d 85 63  fd
44f8 : a5 8e 85 62 a2 90 38 20  62
4500 : 49 bc a9 33 a0 4b 20 28  ad
4508 : ba 20 6b e2 a2 2e a0 4b  be
4510 : 20 d4 bb a5 3f 20 3c bc  9d
4518 : a9 2e a0 4b 20 28 ba 20  d8
4520 : 9b bc a9 30 85 01 a4 3f  fb
4528 : a5 65 91 f7 a9 37 85 01  4f
4530 : e6 3f d0 df ee 20 d0 18  49
4538 : a5 8d 69 04 85 8d 90 02  8a
4540 : e6 8e e6 f8 a5 f8 c9 db  47
4548 : 90 aa a9 00 8d 20 d0 a8  06
4550 : 99 00 4c 99 00 4d 99 00  00
4558 : 4e 99 00 51 99 00 52 c8  12
4560 : d0 ee 85 03 85 04 a5 3f  f7
4568 : 20 3c bc a2 2e a0 4b 20  7f
4570 : d4 bb a9 38 a0 4b 20 a2  bd
4578 : bb a9 2e a0 4b 20 0f bb  11
4580 : 20 9b bc a9 30 85 01 a4  4f
4588 : 3f a5 65 99 00 db a9 37  1a
4590 : 85 01 e6 3f d0 d0 a9 00  72
4598 : 8d 30 d0 a9 3b 8d 11 d0  ad
45a0 : ee ff 03 60 ad d6 4a 49  a8
45a8 : 01 8d d6 4a ad e5 48 49  2c
45b0 : 01 8d e5 48 a2 37 86 01  fa
45b8 : a0 40 ad 00 dd 29 02 49  a5
45c0 : 02 8d 00 dd f0 55 a9 00  a5
45c8 : 99 00 e0 99 40 e1 99 80  47
45d0 : e2 99 c0 e3 99 00 e5 99  90
45d8 : 40 e6 99 80 e7 99 c0 e8  22
45e0 : 99 00 ea 99 40 eb 99 80  32
45e8 : ec 99 c0 ed 99 00 ef 99  1b
45f0 : 40 f0 99 80 f1 99 c0 f2  f3
45f8 : 99 00 f4 99 40 f5 99 80  1d
4600 : f6 99 c0 f7 99 00 f9 99  a7
4608 : 40 fa 99 80 fb 99 c0 fc  c5
4610 : 99 00 fe c8 d0 b2 a2 30  0f
4618 : a6 01 60 99 00 60 99 40  74
4620 : 61 99 80 62 99 c0 63 99  1b
4628 : 00 65 99 40 66 99 80 67  4d
4630 : 99 c0 68 99 00 6a 99 40  b1
4638 : 6b 99 80 6c 99 c0 6d 99  a6
4640 : 00 6f 99 40 70 99 80 71  1f
4648 : 99 c0 72 99 00 74 99 40  9c
4650 : 75 99 80 76 99 c0 77 99  31
4658 : 00 79 99 40 7a 99 80 7b  f1
4660 : 99 c0 7c 99 00 7e c8 d0  64
4668 : b2 a2 30 86 01 60 4c 7e  89
4670 : 47 a9 7d a0 4b 85 bb 84  d8
4678 : bc a9 01 85 b7 a9 60 85  4f
4680 : b9 48 a9 08 85 ba 48 20  58
4688 : d5 f3 68 20 b4 ff 68 20  a2
4690 : 96 ff a9 00 85 90 20 e9  c1
4698 : 46 20 e9 46 20 e9 46 20  dc
46a0 : e9 46 20 e9 46 85 63 20  50
46a8 : e9 46 85 62 a2 90 38 20  32
46b0 : 49 bc 20 df bd a8 b9 00  63
46b8 : 01 f0 06 99 81 4b c8 d0  1d
46c0 : f5 a9 20 99 81 4b c8 84  64
46c8 : f7 20 e9 46 f0 09 a4 f7  ec
46d0 : 99 81 4b e6 f7 d0 f2 a9  ff
46d8 : 0d a4 f7 99 81 4b 20 1d  96
46e0 : 47 a5 cb c9 40 f0 b5 d0  2a
46e8 : f8 20 a5 ff 48 a5 90 d0  ef
46f0 : 02 68 60 68 68 68 20 42  1a
46f8 : f6 a9 0d 20 d2 ff a9 08  ee
4700 : 85 ba 20 b4 ff a9 6f 85  97
4708 : b9 20 96 ff 20 a5 ff 20  e6
4710 : d2 ff c9 0d d0 f6 a9 00  61
4718 : 85 c6 4c ab ff a0 03 b9  0d
4720 : 81 4b c9 22 f0 44 c8 b9  c5
4728 : 81 4b c9 42 f0 3c a0 18  ad
4730 : b9 81 4b c9 53 d0 46 88  9c
4738 : b9 81 4b c9 22 d0 f8 a2  90
4740 : 02 88 b9 81 4b dd 7e 4b  59
4748 : d0 33 88 ca 10 f4 c8 84  02
4750 : f8 a9 3e 20 d2 ff a0 05  6a
4758 : a9 20 20 d2 ff c8 c0 06  c9
4760 : 90 f8 20 6e 47 a9 0d 4c  d1
4768 : d2 ff a0 00 84 f8 b9 81  5c
4770 : 4b 20 d2 ff c8 c4 f8 f0  f8
4778 : 04 c9 0d d0 f1 60 a9 00  87
4780 : 85 02 85 b6 58 a9 00 8d  2c
4788 : 77 02 85 c6 20 f3 47 48  8a
4790 : 20 0a 48 68 d0 0b 20 f3  a2
4798 : 47 d0 06 20 0a 48 4c 85  ec
47a0 : 47 a4 02 84 8d 20 b0 47  76
47a8 : a5 8d 20 0c 48 4c 85 47  29
47b0 : c9 11 f0 23 c9 91 f0 2e  ec
47b8 : c9 0d f0 14 c9 31 90 16  5b
47c0 : c9 39 b0 12 e9 30 85 02  ce
47c8 : a5 8d 20 0c 48 20 0a 48  fc
47d0 : e6 02 68 68 a5 02 60 a5  16
47d8 : 02 c9 07 b0 02 e6 02 a5  41
47e0 : b6 f0 1e c6 b6 60 a5 02  78
47e8 : f0 02 c6 02 a5 b6 d0 11  41
47f0 : e6 b6 60 a0 50 ad 77 02  b2
47f8 : d0 06 ca d0 f8 88 d0 f5  9b
4800 : 60 a0 38 ca d0 fd 88 d0  d8
4808 : fa 60 a5 02 0a a8 b9 a2  ee
4810 : 4b 85 f7 b9 a3 4b 85 f8  ef
4818 : a0 02 b1 f7 49 80 91 f7  f3
4820 : 88 10 f7 60 a4 ae c4 bd  08
4828 : d0 03 4c f8 49 a6 ad a5  78
4830 : ac e4 bc d0 07 c5 bb d0  c7
4838 : 23 4c 84 4a e4 bc f0 1c  1c
4840 : b0 1e 38 a5 bb e5 ac 85  6b
4848 : fb a5 bc e5 ad 85 fc a9  50
4850 : 50 a2 49 8d 3d 49 8e 3e  ca
4858 : 49 4c 73 48 c5 bb 90 e2  ef
4860 : e5 bb 85 fb 8a e5 bc 85  d9
4868 : fc a9 71 a2 49 8d 3d 49  72
4870 : 8e 3e 49 c4 bd b0 14 38  2a
4878 : a5 bd e5 ae 85 8e a9 92  e4
4880 : a0 49 8d 24 49 8c 25 49  cd
4888 : 4c 9a 48 98 e5 bd 85 8e  c6
4890 : a9 c4 a0 49 8d 24 49 8c  25
4898 : 25 49 a4 fb a5 fc d0 04  98
48a0 : c4 8e 90 0a 4a 98 6a 85  2f
48a8 : c1 85 fd 4c d1 48 a5 8e  48
48b0 : 85 fb 4a 85 c1 85 fd 84  bf
48b8 : 8e ad 24 49 ac 3d 49 8d  44
48c0 : 3d 49 8c 24 49 ad 25 49  73
```

Listing 2. (Fortsetzung)

```
48c8 : ac 3e 49 8d 3e 49 8c 25   42
48d0 : 49 a0 00 84 c2 a6 bd 18   82
48d8 : a5 bb 29 f8 7d 00 58 85   08
48e0 : f9 a5 bc 7d 00 56 85 fa   49
48e8 : a5 bb 29 07 aa bd 3d 4b   ba
48f0 : 85 07 b1 f9 05 07 91 f9   67
48f8 : 20 c6 4a b1 f7 05 b6 91   ea
4900 : f7 a5 8e 8d 14 49 a6 fd   41
4908 : f0 45 a4 f7 a2 00 86 f7   f7
4910 : 18 a5 c1 69 00 85 c1 a5   17
4918 : c2 69 00 85 c2 c5 fc f0   70
4920 : 15 90 19 20 ff ff 38 a5   f4
4928 : c1 e5 fb 85 c1 a5 c2 e5   ab
4930 : fc 85 c2 4c 3c 49 a5 c1   51
4938 : c5 fb b0 e7 20 ff ff b1   89
4940 : f7 05 b6 91 f7 a1 f9 05   18
4948 : 07 81 f9 c6 fd d0 c1 60   95
4950 : 46 b6 90 0a 66 b6 98 69   a8
4958 : 08 a8 90 02 e6 f8 06 07   75
4960 : b0 01 60 26 07 a5 f9 e9   c7
4968 : 07 85 f9 90 01 60 c6 fa   e6
4970 : 60 06 b6 90 0a 26 b6 98   71
4978 : e9 07 a8 b0 02 c6 f8 46   ec
4980 : 07 b0 01 60 66 07 a5 f9   55
4988 : 69 08 85 f9 b0 01 60 e6   f8
4990 : fa 60 98 29 07 c9 07 f0   c2
4998 : 0a c8 a5 f9 29 07 f0 16   6a
49a0 : c6 f9 60 98 69 38 a8 a5   d4
49a8 : f8 69 01 85 f8 a5 f9 29   3d
49b0 : 07 f0 03 c6 f9 60 38 a5   98
49b8 : f9 e9 39 85 f9 a5 fa e9   31
49c0 : 01 85 fa 60 98 29 07 f0   1f
49c8 : 0c 88 a5 f9 29 07 c9 07   c1
49d0 : f0 19 e6 f9 60 38 98 e9   44
49d8 : 39 a8 a5 f8 e9 01 85 f8   9c
49e0 : a5 f9 29 07 c9 07 f0 03   4c
49e8 : e6 f9 60 a5 f9 69 38 85   6e
49f0 : f9 a5 fa 69 01 85 fa 60   91
49f8 : a5 bc a6 bb c5 ad d0 02   2d
```

```
4a00 : e4 ac b0 0c a4 ad 85 ad   11
4a08 : 84 bc a4 ac 86 ac 84 bb   00
4a10 : a5 ac 48 a5 ad 48 a5 bb   fd
4a18 : 85 ac a5 bc 85 ad a0 00   3c
4a20 : 20 c6 4a 86 02 a5 f7 85   3f
4a28 : f9 a5 f8 85 fa 68 85 ad   47
4a30 : 68 85 ac 20 c6 4a a5 f7   cf
4a38 : c5 f9 d0 13 a5 f8 e5 fa   40
4a40 : d0 0d bd 45 4b a6 02 3d   1b
4a48 : 4d 4b 11 f7 91 f7 60 bd   54
4a50 : 45 4b 48 38 a5 f9 e5 f7   06
4a58 : aa a5 fa e5 f8 4a 8a 6a   31
4a60 : 4a 4a aa 18 68 11 f7 2c   c4
4a68 : a9 ff 91 f7 a5 f7 69 08   44
4a70 : 85 f7 90 03 e6 f8 18 ca   a1
4a78 : d0 ee a6 02 bd 4d 4b 11   3f
4a80 : f7 91 f7 60 a5 bd c5 ae   06
4a88 : b0 08 a4 ae 85 ae 84 bd   97
4a90 : 38 98 e5 ae 48 a5 ae 29   22
4a98 : 07 a8 a5 ae 29 f8 aa 20   78
4aa0 : c8 4a 68 aa e8 a5 b6 11   b5
4aa8 : f7 91 f7 ca f0 17 c8 c0   2b
4ab0 : 08 90 f2 a5 f7 69 3f 85   44
4ab8 : f7 a5 f8 69 01 85 f8 a5   59
4ac0 : b6 a0 00 f0 e0 60 a6 ae   ed
4ac8 : 18 a5 ac 29 f8 7d 00 58   2f
4ad0 : 85 f7 a5 ad 7d 00 56 85   ac
4ad8 : f8 a5 ac 29 07 aa bd 3d   2a
4ae0 : 4b 85 b6 60 20 fd ae 20   94
4ae8 : eb b7 a5 14 a4 15 85 ac   fd
4af0 : 84 ad 20 c8 4a a0 00 11   38
4af8 : f7 91 f7 60 20 fd ae 20   af
4b00 : eb b7 86 bd a5 14 a4 15   d8
4b08 : 85 bb 84 bc 20 fd ae 20   10
4b10 : eb b7 86 ae a4 ae a6 15   d3
4b18 : a5 14 86 ad 85 ac e4 bc   e9
4b20 : d0 07 c5 bb 20 5c 48 58   13
4b28 : 60 20 3c 48 58 60 00 00   39
4b30 : 00 00 00 7b 0e fa 35 0f   4c
```

```
4b38 : 81 4c cc cc cd 80 40 20   ce
4b40 : 10 08 04 02 01 ff 7f 3f   22
4b48 : 1f 0f 07 03 01 80 c0 e0   ea
4b50 : f0 f8 fc fe ff 02 01 08   ff
4b58 : 04 40 80 02 10 80 10 fe   20
4b60 : df df fb fd fb f7 fd fd   61
4b68 : bf 6d 42 27 43 35 43 1d   78
4b70 : 43 22 43 cf 42 e9 42 03   12
4b78 : 43 ca 42 ca 42 24 2e 33   6f
4b80 : 44 20 20 20 20 20 20 20   a4
4b88 : 20 20 20 20 20 20 20 20   88
4b90 : 20 20 20 20 20 20 20 20   90
4b98 : 20 20 20 20 20 20 20 20   98
4ba0 : 20 20 c9 04 19 05 69 05   2d
4ba8 : b9 05 09 06 59 06 a9 06   5f
4bb0 : f9 06 a9 01 a8 91 2b 20   3b
4bb8 : 33 a5 18 a5 22 69 02 85   f9
4bc0 : 2d a5 23 69 00 85 2e 60   5c
```

Listing 2. (Schluß)

```
Name : 3d.sprt                5bc0 5c02
---------------------------------------
5bc0 : 18 00 00 1c 00 00 0e 00   94
5bc8 : 00 07 00 00 03 80 00 01   82
5bd0 : c0 00 00 e0 00 00 70 00   6e
5bd8 : 00 38 00 00 1c 00 00 0e   d2
5be0 : 00 00 07 00 00 03 80 00   bc
5be8 : 01 c0 00 00 f0 00 00 78   49
5bf0 : 00 00 7e 00 00 3f 00 00   8a
5bf8 : 1f 00 00 1c 00 00 0c 00   cb
5c00 : 01 08 ff ff ff ff 00 00   05
```

Listing 3. »3D-SPRT« – Spritedaten des Grafik-Cursors (bitte mit dem MSE eingeben)

```
0 REM"-MTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTG      <176>
1 REM"-M{7SPACE}3D - MASTER{9SPACE}G         <189>
2 REM"-M{5SPACE}VON J.SCHWARZER{7SPACE}G     <064>
5 REM"- TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT       <071>
10 :                                          <242>
100 SYS 2↑14+9:SYS 2↑14:POKE 3,0:POKE 4,0    <233>
110 POKE 650,128:POKE 53281,0                 <105>
120 POKE 56,64:CLR:LO=1:DW=45                 <046>
130 V=53248:P=π/180                           <157>
140 S=5:M=40:XM=160:YM=100:W=30               <185>
150 BA=2↑14:L=BA+3:IN=BA+6:SE=BA+12:VP=BA+
    15:PL=BA+18:NR=BA+21                      <164>
160 SQ=BA+24:LQ=BA+27:IP=BA+30:DP=BA+33       <233>
170 IL=BA+36:DL=BA+39:DR=BA+42                <083>
180 GOSUB 10850                               <226>
190 P1=20736:P2=20992                         <057>
200 XS=19456:YS=19712:ZS=19968                <011>
210 POKE 191,DW/4                             <033>
220 DIM X(128),Y(128)                         <042>
230 D=256:R=127:RI$=CHR$(13):W$="{DOWN,RIG
    HT,WHITE,SPACE}":H$="{HOME}":C$="{CLR}   <253>
240 CU$="{17DOWN}"                            <075>
250 SS$="-----------------------------------
    ------                                    <162>
260 SP$="{39SPACE}":D$="{DOWN}":U$="{UP}":
    L$="{LEFT}":R$="{RIGHT}"                  <213>
270 PRINT C$:REM -----                        <054>
280 POKE 24568,111:POKE V+39,2                <219>
1000 REM MENUE                                <170>
1010 GOSUB 1240:Z=1:QS=81:QR=87:POKE 53280
     ,0:POKE V+21,0                           <073>
1020 PRINT "{HOME,DOWN,RIGHT,WHITE,CTRL-N,
     CTRL-H}EINGABE-MENUE:                    <073>
1030 PRINT"{3DOWN,RIGHT,SPACE}1{SPACE,RED}
     ........ EINGEBEN ALS 3D-ZEICHNUNG       <073>
1040 PRINT W$"2{SPACE,ORANGE}... EINGEBEN
     ALS KOORDINATENTRIPEL                    <086>
1050 PRINT W$"3{SPACE,YELLOW}... EINGEBEN
     VON ROTATIONSKOERPERN                    <074>
1060 PRINT W$"4{SPACE,GREEN}.............
     .. KOORDINATEN LADEN                     <203>
1070 PRINT W$"5{SPACE,GREY 1}............
     KOORDINATEN SPEICHERN                    <126>
1080 PRINT W$"6{SPACE,BLUE}...............
     ......... DIRECTORY                      <082>
1085 PRINT W$"7{SPACE,LIG.BLUE}...........
     ................... DREHEN               <044>
1090 PRINT"{2RIGHT,WHITE,32SPACE}@@@@         <071>
1100 PRINT"{RIGHT,SPACE}8 ................
     .............{SPACE,RVSON}EXIT           <230>
1110 SYS BA+1646                              <090>
1230 POKE 198,0:ON PEEK(2)GOTO 10000,20000
     ,30000,50010,40010,50100,60020,60000     <222>
1240 POKE V+24,21:POKE V+17,27:POKE 56576,
     3:RETURN                                 <191>
1250 POKE V+24,120:POKE V+17,59:POKE 56576
     ,2:RETURN                                <202>
1260 POKE 1146+Z*80,Q                         <086>
1270 QC=PEEK(55420+Z*80)                      <040>
1280 IF Q=QR THEN QC=1                        <168>
1290 POKE 55418+Z*80,QC:RETURN                <001>
```

Listing 4. Der Basic-Editor »3D.Edit«

```
1300 -----                                   <066>
10000 POKE V+21,0:SYS BA:POKE 56576,2:X=0:
      Y=0:Z=0                                <123>
10010 GOSUB 10820                            <213>
10020 IF LO THEN LO=0:AP=0:AL=0:POKE 3,0:P
      OKE 4,0:GOTO 10140                     <125>
10030 IF(AP OR AL)=0 GOTO 10140              <180>
10040 SYS NR:FOR I=0 TO AL-1                 <021>
10050 X=PEEK(XS+PEEK(P1+I)):X=X+D*(X>R)      <171>
10060 Y=PEEK(YS+PEEK(P1+I)):Y=Y+D*(Y>R)      <015>
10070 Z=PEEK(ZS+PEEK(P1+I)):Z=Z+D*(Z>R)      <115>
10080 GOSUB 10350:XA=XX:YA=YY                <094>
10090 X=PEEK(XS+PEEK(P2+I)):X=X+D*(X>R)      <227>
10100 Y=PEEK(YS+PEEK(P2+I)):Y=Y+D*(Y>R)      <071>
10110 Z=PEEK(ZS+PEEK(P2+I)):Z=Z+D*(Z>R)      <171>
10120 GOSUB 10350:SYS L,XA,YA,XX,YY:NEXT     <110>
10130 SYS IN                                 <136>
10140 XA=160:YA=100:XB=XA:YB=YA:X=0:Y=0:Z=
      0                                      <225>
10150 GOSUB 10850:POKE V+21,1                <037>
10160 GOSUB 60050                            <083>
10170 IF A$<>""GOTO 10220                    <100>
10180 GOSUB 10320                            <222>
10190 GOSUB 60050                            <113>
10200 GOSUB 10320                            <242>
10210 IF A$=""GOTO 10160                     <054>
10220 GOSUB 10490                            <232>
10230 XR=XA:YR=YA:XP=XB:YP=YB                <121>
10240 GOSUB 10320:GOSUB 10250:SYS L,XR,YR,
      XP,YP:ON MM+1 GOTO 10160,10000         <100>
10250 IF Q=1 THEN X1=X:Y1=Y:Z1=Z:GOSUB 103
      40:XA=XX:YA=YY:GOSUB 10850             <180>
10260 IF A$="*"THEN GOSUB 10410:GOSUB 1034
      0:GOSUB 10850                          <051>
10270 IF A$=RI$THEN GOSUB 10320:GOSUB 1068
      0:GOSUB 10410:GOTO 10290               <194>
10280 RETURN                                 <178>
10290 XL=X1:YL=Y1:ZL=Z1:X1=X2:Y1=Y2:Z1=Z2:
      X=X1:Y=Y1:Z=Z1:XA=XB:YA=YB:RETURN      <186>
10300 -----                                  <176>
10310 REM LINIE ZIEHEN                       <231>
10320 SYS L,XA,YA,XB,YB:RETURN               <115>
10330 -----                                  <206>
10340 REM 3D NACH 2D KOORDINATEN             <220>
10350 YT=Y/1.6                               <184>
10360 YY=YM-Z-YT*SI                          <179>
10370 XX=XM+X+YT*CO                          <140>
10380 RETURN                                 <022>
10390 -----                                  <010>
10400 REM 2.PUNKT SUCHEN                     <198>
10410 XX=X1:YY=Y1:ZZ=Z1                      <162>
10420 X1=X2:Y1=Y2:Z1=Z2                      <226>
10430 X2=XX:Y2=YY:Z2=ZZ                      <020>
10440 XX=XA:YY=YA:XA=XB                      <249>
10450 YA=YB:XB=XX:YB=YY                      <008>
10460 X=X1:Y=Y1:Z=Z1                         <041>
10470 RETURN                                 <112>
10480 -----                                  <100>
10490 Q=0:MM=0                               <121>
10500 IF A$="X"OR A$=R$THEN X=X+S:Q=1:IF X
      >M THEN X=M                            <042>
10510 IF A$="X"OR A$=L$THEN X=X-S:Q=1:IF X
      <-M THEN X=-M                          <065>
10520 IF A$="Y"THEN Y=Y+S:Q=1:IF Y>M THEN
      Y=M                                    <129>
10530 IF A$="Y"THEN Y=Y-S:Q=1:IF Y<-M THEN
       Y=-M                                  <006>
10540 IF A$="Z"OR A$=U$THEN Z=Z+S:Q=1:IF Z
      >M THEN Z=M                            <206>
10550 IF A$="Z"OR A$=D$THEN Z=Z-S:Q=1:IF Z
      <-M THEN Z=-M                          <133>
10560 IF A$=C$THEN LO=1:GOTO 10000           <122>
10570 IF A$=H$THEN X=0:Y=0:Z=0:Q=1           <014>
10580 IF A$="+"THEN IF DW<90-S THEN DW=DW+
      S:POKE 191,DW/4:MM=1:GOTO 10820        <016>
10590 IF A$="-"THEN IF DW>S THEN DW=DW-S:P
      OKE 191,DW/4:MM=1:GOTO 10820           <180>
10600 IF A$="K"GOTO 10870                    <166>
10610 IF A$="P"THEN X1=X:Y1=Y:Z1=Z:XB=XA:Y
      B=YA:X2=X:Y2=Y:Z2=Z                    <106>
10620 IF A$="L"THEN X=XL:Y=YL:Z=ZL:Q=1       <071>
10630 IF A$="A"THEN GOSUB 10950              <143>
10640 IF A$>"/"AND A$<":"THEN S=ASC(A$)-48
      :IF S=0 THEN S=10                      <087>
```

```
10650 IF A$="←"OR A$="M"GOTO 60030           <222>
10660 RETURN                                 <048>
10670 -----                                  <036>
10680 REM KOORDINATEN SPEICHERN              <049>
10690 IF AP=255 OR AL=255 THEN RETURN        <230>
10700 X=X1:Y=Y1:Z=Z1:GOSUB 10760             <250>
10710 PP=P                                   <032>
10720 X=X2:Y=Y2:Z=Z2:GOSUB 10760             <179>
10730 SYS VP,PP,P                            <250>
10740 AL=PEEK(4):RETURN                      <043>
10750 -----                                  <116>
10760 SYS SE,X-256*(X<0),Y-256*(Y<0),Z-256
      *(Z<0)                                 <011>
10770 AP=PEEK(3):P=PEEK(782):RETURN          <128>
10780 -----                                  <148>
10790 POKE V+21,0                            <012>
10800 POKE 191,DW/4:SYS BA:SYS D3:POKE 565
      76,2                                   <047>
10810 POKE V+21,1                            <033>
10820 CO=COS(P*DW):SI=SIN(P*DW)              <018>
10830 RETURN                                 <220>
10840 -----                                  <208>
10850 POKE V,XA+1 AND 255:POKE V+1,YA+30:R
      ETURN                                  <013>
10860 -----                                  <228>
10870 XD=X:YD=Y:ZD=Z                         <162>
10880 X=-M:Y=0:Z=0:GOSUB 10340:XQ=XX:YQ=YY   <194>
10890 X=M:GOSUB 10340:SYS L,XX,YY,XQ,YQ      <237>
10900 X=0:Y=-M:GOSUB 10340:XQ=XX:YQ=YY       <139>
10910 Y=M:GOSUB 10340:SYS L,XX,YY,XQ,YQ      <005>
10920 Y=0:Z=-M:GOSUB 10340:XQ=XX:YQ=YY       <227>
10930 Z=M:GOSUB 10340:SYS L,XX,YY,XQ,YQ      <029>
10940 X=XD:Y=YD:Z=ZD:RETURN                  <157>
10950 PRINT"{CLR,DOWN,RIGHT}KOORDINATEN AN
      ZEIGEN:                                <106>
10960 PRINT"{DOWN,2RIGHT}X="X                <164>
10970 PRINT"{DOWN,2RIGHT}Y="Y                <050>
10980 PRINT"{DOWN,2RIGHT}Z="Z                <193>
10990 POKE V+21,0:GOSUB 1240                 <188>
11000 POKE 198,0:WAIT 198,1:POKE 198,0       <179>
11010 POKE V+21,1:GOTO 1250                  <227>
11020 -----                                  <134>
20000 ZE=0:GOSUB 1240:ZZ=1280                <178>
20010 GOSUB 1240:ZZ=1280:PRINT C$            <186>
20020 ZU=0:IF AP=0 GOTO 20050                <042>
20030 IF AP>ZE+15 THEN ZU=15:GOTO 20050      <003>
20040 ZU=(AP-1)AND 15:IF AP=0 THEN ZU=-1     <133>
20050 Y1=0                                   <013>
20060 GOSUB 21120                            <156>
20070 FOR I=ZE TO ZE+ZU                      <032>
20080 GOSUB 21190                            <114>
20090 NEXT:X1=0:XA=0:Y1=ZU:IF Q=5 THEN Y1=
      T:IF Y1>ZU THEN Y1=ZU                  <145>
20100 PRINT SS$                              <217>
20110 :                                      <018>
20120 GOSUB 60050                            <137>
20130 IF A$<>""GOTO 20190                    <111>
20140 GOSUB 21080                            <078>
20150 GOSUB 60050                            <167>
20160 GOSUB 21080                            <098>
20170 IF A$=""GOTO 20120                     <046>
20180 :                                      <088>
20190 GOSUB 21080                            <128>
20200 Q=0                                    <097>
20210 GOSUB 20260                            <067>
20220 ON Q GOTO 20000,20120,20050,21280,20
      010                                    <009>
20230 POKE A,C                               <054>
20240 GOTO 20120                             <050>
20250 :                                      <160>
20260 IF A$=D$GOTO 20870                     <156>
20270 IF A$=U$GOTO 20940                     <194>
20280 IF A$=R$GOTO 20980                     <075>
20290 IF A$=L$GOTO 21030                     <064>
20300 IF A$=C$THEN AP=0:POKE 3,0:A$=H$:Q=1   <204>
20310 IF A$=H$THEN X1=0:Y1=0                 <226>
20320 IF A$=CHR$(148)AND AP<>255 THEN SYS
      IP,ZE+Y1:AP=PEEK(3):Q=5:T=Y1:RETURN    <178>
20330 IF A$=CHR$(20)THEN IF AP<>0 THEN SYS
       DP,ZE+Y1:AP=PEEK(3):Q=5:T=Y1:IF AP
      AND 15=15 GOTO 20543                   <074>
```

Listing 4. (Fortsetzung)

```
20340 IF A$>="0"AND A$<":"GOTO 20680          <012>
20350 IF A$="+"OR A$=" "OR A$="-"GOTO 2079
      0                                       <249>
20360 IF A$="←"OR A$="M"THEN POKE 3,AP:GOT
      O 60030                                 <014>
20370 IF A$="{F7}"THEN Q=4:POKE 3,AP:RETUR
      N                                       <199>
20380 IF A$<>RI$THEN RETURN                   <098>
20390 :                                       <044>
20400 REM RETURN TASTE                        <252>
20410 X1=0:XA=0:POKE A,C                      <177>
20420 IF Y1<ZU THEN Y1=Y1+1:RETURN            <041>
20430 IF ZU<>15 GOTO 20500                    <049>
20440 IF ZE<240 THEN ZE=ZE+16                 <177>
20450 PRINT"{CLR}":Y1=0                       <004>
20460 IF ZE=AP THEN ZU=0:Q=3:AP=AP+1:POKE
      3,AP:RETURN                             <023>
20470 IF AP<ZE+17 THEN ZU=AP-1 AND 15         <224>
20480 Q=3:RETURN                              <151>
20490 :                                       <150>
20500 ZU=ZU+1:AP=AP+1:POKE 3,AP:IF AP=1 TH
      EN ZU=0:Q=3:RETURN                      <154>
20505 GOSUB 21120                             <093>
20510 I=AP-1:PRINT D$SP$U$                    <205>
20520 GOSUB 21190                             <046>
20530 Y1=Y1+1                                 <232>
20540 PRINT SS$:Q=2:RETURN                    <061>
20541 -----                                   <001>
20542 REM DELETE POINT                        <112>
20543 IF ZU=0 THEN IF ZE>15 AND AP=ZE THEN
       ZE=ZE-16:T=15                          <019>
20544 RETURN                                  <026>
20550 -----                                   <010>
20560 REM ZEILE UEBERNEHMEN                   <183>
20570 Q=ZZ+Y1*40:X$="":Y$=X$:Z$=X$            <249>
20580 FOR I=0 TO 2                            <199>
20590 X$=X$+CHR$(PEEK(Q-1+I))                 <241>
20600 Y$=Y$+CHR$(PEEK(Q+7+I))                 <128>
20610 Z$=Z$+CHR$(PEEK(Q+15+I)):NEXT           <079>
20620 Q=ZE+Y1                                 <218>
20630 X=VAL(X$):POKE XS+Q,X-D*(X<0)           <069>
20640 Y=VAL(Y$):POKE YS+Q,Y-D*(Y<0)           <188>
20650 Z=VAL(Z$):POKE ZS+Q,Z-D*(Z<0)           <051>
20660 Q=0:RETURN                              <027>
20670 -----                                   <130>
20680 REM ZAHL EINGEBEN                       <213>
20690 X2=ZZ+X1+Y1*40                          <192>
20700 I=ZZ+(X1 AND 24)+Y1*40-1                <102>
20710 X$="":FOR N=I TO I+2                    <131>
20720 T=PEEK(N)AND 127:IF N=X2 THEN T=ASC(
      A$)                                     <127>
20730 X$=X$+CHR$(T):NEXT                      <084>
20740 T=VAL(X$):IF ABS(T)>M THEN RETURN       <153>
20750 C=ASC(A$):T=T-D*(T<0)                   <122>
20760 POKE XS+((X1 AND 24)/8)*D+ZE+Y1,T       <223>
20770 GOTO 20990                              <011>
20780 -----                                   <242>
20790 IF A$="+"THEN A$=" "                    <083>
20800 X2=ZZ+(X1 AND 24)-1+Y1*40               <230>
20810 POKE X2,ASC(A$)                         <166>
20820 X2=XS+((X1 AND 24)/8)*D+ZE+Y1           <161>
20830 T=PEEK(X2):T=-T-D*(T>R)                 <061>
20840 T=T-D*(T<0):POKE X2,T:RETURN            <174>
20850 -----                                   <056>
20860 REM DOWN                                <239>
20870 IF Y1<ZU THEN Y1=Y1+1:Q=0:RETURN        <051>
20880 IF ZU<>15 OR AP<ZE+17 THEN RETURN       <251>
20890 IF ZE<240 THEN ZE=ZE+16                 <119>
20900 ZU=15:IF AP<ZE+16 THEN ZU=AP-1 AND 1
      5                                       <095>
20910 Q=3:Y1=0:PRINT C$:RETURN                <229>
20920 -----                                   <126>
20930 REM UP                                  <081>
20940 IF Y1>0 THEN Y1=Y1-1:RETURN             <092>
20950 IF ZE>0 THEN ZE=ZE-16:Q=3:ZU=15:PRIN
      T C$                                    <115>
20960 RETURN                                  <188>
20970 -----                                   <176>
20980 REM RIGHT                               <030>
20990 IF X1=1 OR X1=9 THEN XA=XA+1:X1=XA*8
      :RETURN                                 <057>
21000 IF X1<17 THEN X1=X1+1                   <169>
21010 RETURN                                  <240>
21020 -----                                   <228>
21030 REM LEFT                                <155>
21040 IF X1=8 OR X1=16 THEN XA=XA-1:X1=XA*
      8:RETURN                                <178>
21050 IF X1>0 THEN X1=X1-1                    <016>
21060 RETURN                                  <034>
21070 -----                                   <022>
21080 A=ZZ+X1+Y1*40:C=PEEK(A)                 <118>
21090 POKE A,(C OR 128)-(C AND 128)           <063>
21100 RETURN                                  <074>
21110 -----                                   <062>
21120 PRINT"{GREY 1}                          <020>
21130 PRINT"{HOME,CTRL-N,RIGHT}KOORDINATEN
      TRIPEL EINGEBEN:{DOWN}                  <133>
21140 PRINT"{RIGHT,SPACE}ANZAHL DER PUNKTE
      :"AP                                    <214>
21150 PRINT"{RIGHT,SPACE}ANZAHL DER LINIEN
      :"AL                                    <223>
21160 PRINT D$SS$                             <102>
21170 PRINT U$LEFT$(CU$,Y1):RETURN            <144>
21180 -----                                   <132>
21190 REM INHALT DES PUNKTESPEICHERS          <027>
21200 REM ANZEIGEN                            <234>
21210 PRINT"NR."I+1"{LEFT,SPACE}"TAB(8)       <132>
21220 X=PEEK(XS+I):X=X+D*(X>R)                <019>
21230 Y=PEEK(YS+I):Y=Y+D*(Y>R)                <185>
21240 Z=PEEK(ZS+I):Z=Z+D*(Z>R)                <095>
21250 PRINT"==>{RVOFF,2SPACE}X="X"{LEFT,SP
      ACE}"TAB(21)"Y="Y"{LEFT,SPACE}"TAB(2
      9)"Z="Z"{LEFT,2SPACE}"                  <166>
21260 RETURN                                  <236>
21270 -----                                   <224>
21280 POKE 3,AP:ZE=0                          <242>
21290 ZU=15:IF AL<ZE+16 THEN ZU=AL-1 AND 1
      5                                       <216>
21300 IF AL=0 THEN ZU=0                       <090>
21310 ZZ=1285                                 <077>
21320 Y1=0:X=0:PRINT"{CLR,YELLOW}":GOSUB 2
      1130                                    <004>
21330 FOR I=ZE TO ZE+ZU                       <022>
21340 GOSUB 22140                             <055>
21350 NEXT:Y=ZU:IF Q=2 THEN Y=T:IF Y>ZU TH
      EN Y=ZU                                 <034>
21360 PRINT SS$                               <207>
21370 GOSUB 60050                             <117>
21380 GOSUB 21520                             <078>
21390 IF A$<>""GOTO 21450                     <028>
21400 GOSUB 60050                             <147>
21410 IF A$<>""GOTO 21450                     <048>
21420 GOSUB 21520                             <118>
21430 GOTO 21370                              <113>
21440 :                                       <078>
21450 Q=0                                     <077>
21460 GOSUB 21570                             <223>
21470 ON Q GOTO 21320,21290,20000             <131>
21480 POKE K,J                                <081>
21490 GOTO 21370                              <173>
21500 :                                       <138>
21510 REM INVERTIEREN                         <008>
21520 K=ZZ+X+Y*40                             <242>
21530 J=PEEK(K)                               <023>
21540 POKE K,(J OR 128)-(J AND 128)           <217>
21550 RETURN                                  <016>
21560 :                                       <200>
21570 IF A$=D$GOTO 21800                      <088>
21580 IF A$=U$GOTO 21860                      <006>
21590 IF A$=R$GOTO 21910                      <007>
21600 IF A$=L$GOTO 21960                      <224>
21610 IF A$=H$GOTO 21990                      <025>
21611 IF A$=C$THEN AL=0:POKE 4,0:Q=2:RETUR
      N                                       <065>
21620 IF A$=CHR$(148)AND AL<>255 THEN SYS
      IL,ZE+Y:AL=PEEK(4):Q=2:T=Y:RETURN       <189>
21630 IF A$=CHR$(20)AND AL<>0 THEN SYS DL,
      ZE+Y:AL=PEEK(4):Q=2:T=Y:IF AL AND 15
      =15 THEN 21791                          <009>
21640 IF A$="{F7}"THEN Q=3:POKE 4,AL:RETUR
      N                                       <040>
21650 IF A$>"/"AND A$<":"GOTO 22010           <029>
21660 IF A$="M"OR A$="←"THEN POKE 4,AL:GOT
      O 60030                                 <042>
21670 IF A$<>RI$THEN RETURN                   <118>
21680 :                                       <064>
21690 POKE K,J:X=0                            <241>
21700 IF Y<ZU THEN Y=Y+1:RETURN               <135>
```

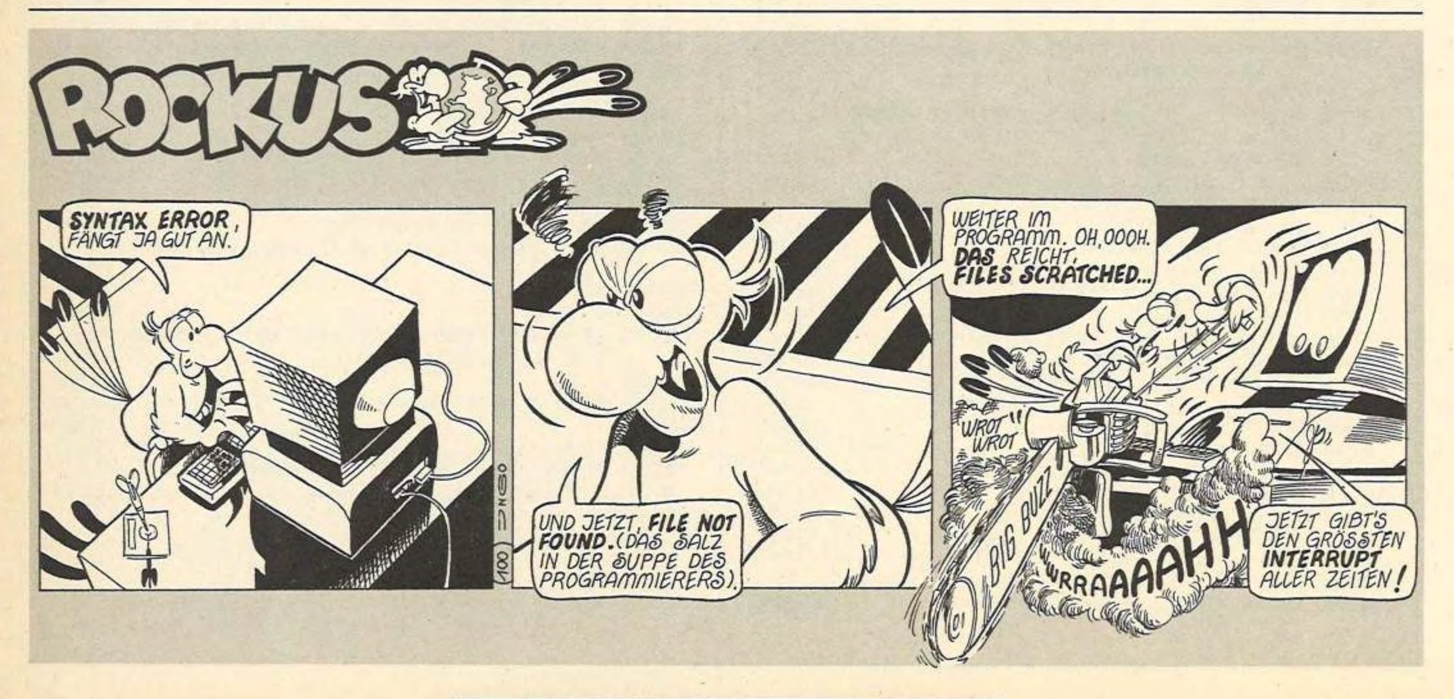
```
21710 AL=AL+1:POKE 4,AL                          <057>
21720 IF ZU=15 GOTO 21780                        <004>
21730 ZU=ZU+1:Y1=ZU:IF AL=1 THEN ZU=0:Y1=0
      :GOTO 21750                                <107>
21740 PRINT U$SP$                                <118>
21750 GOSUB 21130                                <130>
21760 I=AL-1:GOSUB 22140:Y=ZU                    <196>
21770 PRINT SS$:RETURN                           <066>
21780 IF ZE<240 THEN ZE=ZE+16                    <249>
21790 ZU=0:Q=1:RETURN                            <178>
21791 REM DELETE LINE                            <073>
21792 IF ZU=0 THEN IF ZE>15 AND AL=ZE THEN
      ZE=ZE-16:T=15                              <228>
21793 RETURN                                     <005>
21800 REM DOWN                                   <163>
21810 IF Y<ZU THEN Y=Y+1:RETURN                  <247>
21820 IF ZU<>15 OR AL<ZE+17 THEN RETURN          <143>
21830 IF ZE<240 THEN ZE=ZE+16                    <043>
21840 ZU=15:IF AL<ZE+16 THEN ZU=AL-1 AND 1
      5                                          <002>
21850 Q=1:RETURN                                 <219>
21860 REM UP                                     <251>
21870 IF Y>0 THEN Y=Y-1:RETURN                   <213>
21880 IF ZE=0 THEN RETURN                        <057>
21890 ZU=15:IF ZE>0 THEN ZE=ZE-16                <187>
21900 Q=1:RETURN                                 <013>
21910 REM RIGHT                                  <200>
21920 IF X<14 THEN X=X+1:IF X=3 THEN X=12        <171>
21930 IF PEEK(ZZ+X+Y*40)<>32 THEN RETURN         <055>
21940 IF X>12 THEN X=X-1:RETURN                  <102>
21950 GOTO 21920                                 <253>
21960 REM LEFT                                   <067>
21970 IF X>0 THEN X=X-1:IF X=11 THEN X=0         <226>
21980 RETURN                                     <192>
21990 REM HOME                                   <254>
22000 Y=0:X=0:RETURN                             <043>
22010 REM ZAHL EINGEBEN                          <017>
22020 Z=ZZ+Y*40:IF X>11 THEN Z=Z+12              <061>
22030 X$="":FOR I=Z TO Z+2                       <087>
22040 X$=X$+CHR$(PEEK(I)AND 127):NEXT            <182>
22050 Z=X+1:IF X>11 THEN Z=Z-12                  <149>
22060 X$=LEFT$(X$,Z-1)+A$+MID$(X$,Z+1)           <110>
22070 Z=VAL(X$):IF Z>AP OR Z=0 THEN RETURN       <089>
22080 QQ=P1+ZE+Y:IF X>11 THEN QQ=QQ+D            <063>
22090 POKE QQ,Z-1                                <020>
22100 J=ASC(A$):IF X=14 THEN RETURN              <221>
22110 IF X=2 THEN X=12:RETURN                    <021>
22120 X=X+1:RETURN                               <219>
22130 -----                                      <066>
22140 PRINT"NR.:"I+1 TAB(9)"==>";                <147>
22150 PA=PEEK(P1+I)+1:IF PA>AP THEN PA=1:P
      OKE P1+I,1                                 <087>
22160 PB=PEEK(P2+I)+1:IF PB>AP THEN PB=1:P
      OKE P2+I,1                                 <013>
22170 PRINT" VON NR."PA"{LEFT,2SPACE}"TAB(
      25)"BIS NR."PB"{LEFT,3SPACE}"              <022>
22180 RETURN                                     <138>
22190 -----                                      <126>
30000 G$="{HOME,DOWN,RIGHT,CTRL-N,WHITE}RO
      TATIONSKOERPER:"                           <213>
30010 PRINT"{CLR}"G$                             <251>
30020 A$="N":INPUT"{DOWN,RIGHT}ALTE KOORDI
      NATEN LOESCHEN ? N{3LEFT}";A$:IF A$=
      "N"GOTO 30040                              <017>
30030 POKE 3,0:POKE 4,0:AP=0:AL=0                <212>
30040 POKE V+21,0:SYS BA:POKE 56576,2:ZA=0
      :LO=0                                      <070>
30050 X=0:Y=0:X1=X:Y1=Y                          <052>
30060 XA=XM-M*2:YA=YM+M/2:YB=YA                  <001>
30070 FOR I=0 TO 4*M STEP 4                      <177>
30080 H=6:IF I/5=INT(I/5)THEN H=12               <021>
30090 SYS L,XA+I,YA+1,XA+I,YA+H:NEXT             <191>
30100 XA=XM:XB=XA:GOSUB 10850:POKE V+21,1        <142>
30110 FOR I=XM-M*2 TO XM+M*2 STEP 10             <137>
30120 FOR J=YA-M*2 TO YA STEP 10                 <231>
30130 SYS PL,I,J:NEXT J,I                        <078>
30140 GOSUB 60050                                <251>
30150 IF A$<>""GOTO 30210                        <244>
30160 GOSUB 10320                                <134>
30170 GOSUB 60050                                <025>
30180 GOSUB 10320                                <154>
30190 IF A$=""GOTO 30140                         <194>
30200 :                                          <202>
30210 XR=XA:YR=YA:XP=XB:YP=YB:Q=4                <180>
30220 GOSUB 10320                                <196>
30230 GOSUB 30280                                <064>
30240 SYS L,XR,YR,XP,YP                          <089>
30250 IF Q=1 THEN GOSUB 30470:XA=XX:YA=YY:
      GOSUB 10850:Q=4                            <213>
30260 ON Q-1 GOTO 30040,30490,30140              <042>
30270 POKE 53280,1:STOP                          <202>
30280 IF A$="X"OR A$=R$THEN X=X+S:Q=1:IF X
      >M THEN X=M                                <008>
30290 IF A$="X"OR A$=L$THEN X=X-S:Q=1:IF X
      <-M THEN X=-M                              <031>
30300 IF A$="Y"OR A$=U$THEN Y=Y+S:Q=1:IF Y
      >M THEN Y=M                                <137>
30310 IF A$="Y"OR A$=D$THEN Y=Y-S:Q=1:IF Y
      <0 THEN Y=0                                <252>
30320 IF A$="L"THEN X=X2:Y=Y2:Q=1                <250>
30330 IF A$="P"THEN X1=X:Y1=Y:XB=XA:YB=YA        <244>
30340 IF A$=C$THEN Q=2                           <198>
30350 IF A$=H$THEN X=0:Y=0:Q=1                   <209>
30360 IF A$="*"GOTO 30450                        <104>
30370 IF A$=" "THEN Q=3:S=5:RETURN               <077>
30380 IF A$="M"OR A$="←"GOTO 60030               <005>
30390 IF A$>"/"AND A$<":"THEN S=ASC(A$)-48
      :IF S=0 THEN S=10                          <023>
30400 IF A$<>RI$THEN RETURN                      <212>
30410 :                                          <158>
30420 X(ZA)=X1:Y(ZA)=Y1:ZA=ZA+1:X(ZA)=X:Y(
      ZA)=Y:ZA=ZA+1:X2=X1:Y2=Y1                  <126>
30430 X1=X:Y1=Y:GOSUB 30470:GOSUB 10320:XB
      =XX:YB=YY:RETURN                           <173>
30440 :                                          <188>
30450 Q=1:XX=X:YY=Y:X=X1:Y=Y1:X1=XX:Y1=YY:
      XB=XA:YB=YA:RETURN                         <073>
30460 :                                          <208>
30470 XX=XM+X*2:YY=YM+M/2-Y*2:RETURN             <191>
30480 :                                          <230>
30490 GOSUB 1240                                 <124>
30500 PRINT"{CLR}":POKE V+21,0                   <035>
30510 PRINT G$:POKE 19,64                        <204>
30520 INPUT"{DOWN,RIGHT}VON 0 GRAD - 360 G
      RAD{2SPACE,10LEFT}";GR$:GR=INT(ABS(V
      AL(GR$)))                                  <107>
30530 PRINT:IF GR>360 GOTO 30500                 <236>
30540 POKE 19,0                                  <069>
30550 INPUT"{DOWN,RIGHT}ANZAHL DER FACETTE
      N:";AN                                     <034>
30560 IF AN<3 THEN PRINT"{2UP}";:GOTO 3055
      0                                          <028>
30570 INPUT"{DOWN,RIGHT}ALLES OK.";A$            <141>
30580 IF A$<>"N"GOTO 30630                       <205>
30590 INPUT"{DOWN,RIGHT}NEU ZEICHNEN";A$         <064>
30600 IF A$="J"GOTO 30000                        <138>
30610 GOTO 30500                                 <012>
30620 :                                          <114>
30630 AP=PEEK(3):AL=PEEK(4):N=AL                 <244>
30640 A$="{DOWN,RIGHT}ANZAHL DER FACETTEN
      ZU GROSS !!"                               <047>
30650 PRINT"{DOWN,RIGHT}1. FACETTEN{DOWN}        <013>
30660 FOR I=0 TO(GR-1)*P STEP(GR/AN)*P           <218>
30670 FOR J=0 TO ZA-1 STEP 2:PRINT N-AL+1"
      {UP}"                                      <186>
30680 FOR K=0 TO 1:T=K+J                         <017>
30690 X=X(T)-D*(X(T)<0)                          <009>
30700 Y=INT(Y(T)*COS(I)):Y=Y-D*(Y<0)             <062>
30710 Z=INT(Y(T)*SIN(I)):Z=Z-D*(Z<0)             <095>
30720 SYS SE,X,Y,Z                               <035>
30730 POKE P1+N+K*D,PEEK(782)                    <222>
30740 NEXT K:N=N+1:NEXT J:IF I=0 THEN F=N-
      AL                                         <204>
30750 NEXT I                                     <098>
30760 :                                          <000>
30770 IF N>255 THEN PRINT A$:POKE 3,AP:POK
      E 4,AL:GOTO 30510                          <099>
30780 :                                          <020>
30790 POKE 4,N:AP=PEEK(3):PRINT"{2UP}"           <061>
30800 :                                          <040>
30810 T=N-F-1-AL                                 <222>
30820 PRINT"{DOWN,RIGHT}2. VERBINDUNGEN 1        <107>
30830 FOR I=AL TO T+AL STEP F                    <216>
```

```
30840 FOR J=0 TO F-1                              <074>
30850 PA=PEEK(P1+I+J):PB=PEEK(P1+I+J+F)           <009>
30860 IF PA=PB GOTO 30880                         <196>
30870 SYS VP,PA,PB:IF PEEK(4)=255 THEN N=2
      56:GOTO 30770                               <125>
30880 NEXT J,I                                    <215>
30890 IF GR<>360 GOTO 30950                       <134>
30900 FOR K=0 TO F-1                              <142>
30910 PA=PEEK(P1+I+K):PB=PEEK(P1+K+AL)            <080>
30920 IF PA=PB GOTO 30940                         <200>
30930 SYS VP,PA,PB:IF PEEK(4)=255 THEN N=2
      56:GOTO 30770                               <185>
30940 NEXT K                                      <048>
30950 PRINT"{DOWN,RIGHT}3. VERBINDUNGEN 2         <061>
30960 FOR I=AL TO T+AL STEP F                     <090>
30970 FOR J=0 TO F-1                              <204>
30980 PA=PEEK(P2+I+J):PB=PEEK(P2+I+J+F)           <174>
30990 IF PA=PB GOTO 31010                         <156>
31000 SYS VP,PA,PB:IF PEEK(4)=255 THEN N=2
      56:GOTO 30770                               <001>
31010 NEXT J,I                                    <091>
31020 IF GR<>360 GOTO 60030                       <128>
31030 FOR K=0 TO F-1                              <018>
31040 PA=PEEK(P2+I+K):PB=PEEK(P2+K+AL)            <245>
31050 IF PA=PB GOTO 31070                         <056>
31060 SYS VP,PA,PB:IF PEEK(4)=255 THEN N=2
      56:GOTO 30770                               <061>
31070 NEXT K                                      <180>
31080 :                                           <066>
31090 GOTO 60030                                  <040>
40000 -SPEICHERN-                                 <168>
40010 GOSUB 50040                                 <144>
40020 OPEN 1,8,1,B$                               <064>
40030 PRINT#1,CHR$(PEEK(3));                      <213>
40040 PRINT#1,CHR$(PEEK(4));                      <225>
40050 FOR I=0 TO PEEK(3)-1                        <088>
40060 PRINT#1,CHR$(PEEK(XS+I));                   <223>
40070 PRINT#1,CHR$(PEEK(YS+I));                   <235>
40080 PRINT#1,CHR$(PEEK(ZS+I));                   <247>
40090 NEXT                                        <221>
40100 FOR I=0 TO PEEK(4)-1                        <140>
40110 PRINT#1,CHR$(PEEK(P1+I));                   <120>
40120 PRINT#1,CHR$(PEEK(P2+I));                   <134>
40130 NEXT                                        <005>
40140 CLOSE 1:GOTO 60030                          <223>
50000 -LADEN-                                     <166>
50010 IF PEEK(1023)=0 THEN SYS BA:GOSUB 12
      40                                          <050>
50015 GOSUB 50040:LO=0                            <154>
50020 SYS LO,1,8,0,B$:CLOSE 1:GOTO 60030          <099>
50030 -FI LEN AME-                                <115>
50040 B$="":INPUT"{DOWN,RIGHT}FILENAME";B$
      :IF B$="←"OR B$="M"OR B$=""GOTO 6003
      0                                           <087>
50050 B$=LEFT$(B$,13)+".3D,S":RETURN              <018>
50060 -DIREC TO RY-                               <189>
50100 PRINT:SYS BA+1649                           <126>
50110 WAIT 198,1:POKE 198,0:GOTO 60030            <235>
59999 -EXIT-                                      <164>
60000 END                                         <055>
60010 -----                                       <099>
60020 IF(AP OR AL)<>0 THEN PRINT C$:SYS BA
      :SYS DR                                     <051>
60030 POKE V+21,0:POKE 198,0:CLR:AP=PEEK(3
      ):AL=PEEK(4):DW=PEEK(191)*4:GOTO 130        <158>
60040 -----                                       <129>
60050 FOR I=0 TO W:GET A$:IF A$=""THEN NEX
      T                                           <184>
60060 RETURN                                      <171>
```

Listing 4. »3D-EDIT« (Schluß)

```
100 POKE 3,0:POKE 4,0:SYS 2↑14                    <063>
130 V=53248:P=π/40:D=256:M=40                     <110>
150 BA=2↑14:L=BA+3:IN=BA+6:SE=BA+12:VP=BA+
    15:PL=BA+18:NR=BA+21                          <164>
160 SQ=BA+24:LQ=BA+27:IP=BA+30:DP=BA+33           <233>
170 IL=BA+36:DL=BA+39:DR=BA+42                    <083>
190 P1=20736:P2=20992                             <057>
200 XS=19456:YS=19712:ZS=19968                    <011>
210 SX=11:SY=10                                   <246>
1000 MM=M/2                                       <229>
1010 FOR I=-M TO M STEP INT(2*M/(SY-1)):FO
     R J=-M TO M STEP INT(2*M/(SX-1))             <219>
1020 Z=INT(SIN(J*P)*MM+SIN(I*P)*MM)               <222>
1030 Z=Z-D*(Z<0)                                  <074>
1040 SYS SE,J-D*(J<0),I-D*(I<0),Z                 <075>
1050 NEXT J,I                                     <105>
1060 :                                            <020>
1070 :                                            <030>
1080 :                                            <040>
2000 FOR I=0 TO SY-1                              <165>
2010 FOR J=0 TO SX-2                              <186>
2020 SYS VP,J+I*SX,J+I*SX+1                       <180>
2030 NEXT J,I                                     <067>
2040 :                                            <238>
2100 FOR I=0 TO SY-2                              <015>
2110 FOR J=0 TO SX-1                              <028>
2120 SYS VP,J+I*SX,J+(I+1)*SX                     <100>
2130 NEXT J,I                                     <169>
2140 :                                            <084>
3000 SYS BA:SYS DR                                <208>
```

Listing 5. Demo-Programm zur Grafik-Animation



So werden Intros programmiert

Wer kennt sie nicht, die fantastischen Intros und Demos, die vor allem Cracker immer wieder abliefern. Der gewöhnliche Anwender glaubt an Zauberei, wenn er sich das bunte Treiben auf dem Bildschirm ansieht. Wir wollen ein wenig hinter die Kulissen solcher Programme blicken.

In unserem kleinen Kurs werden Sie in einige der Geheimnisse der Intro-Programmierung eingeweiht. Für Einsteiger ist dieser Artikel weniger geeignet, Sie sollten bereits fundierte Kenntnisse in Maschinensprache besitzen und sich mit der Grafik-Programmierung und der dazugehörigen Fachsprache auskennen.

Wir können und wollen hier keineswegs Patentrezepte zur Intro-Programmierung mitteilen. Es geht vielmehr darum, Ihnen Anregungen zu geben, damit Sie anschließend selbst loslegen können. Unsere Betrachtungen beziehen sich auf ein fertiges Intro, das Sie als Listing 1 abgedruckt finden. Es wurde von Dirk Swienty programmiert, und gemeinsam wollen wir es ein wenig zerlegen.

Bitte geben Sie es zunächst einmal mit dem MSE (siehe Seite 159) ein. Dieses Listing ist zwar nicht ganz kurz; das, was es leistet, kann sich jedoch sehen lassen. Damit Sie nicht mehr als notwendig abtippen müssen, haben wir das Listing außerdem noch gepackt. So schrumpft es von 43 auf 21 Blocks. Laden Sie das File mit dem Befehl

```
LOAD"64'ER INTRO",8
```

und starten Sie es mit

```
RUN
```

Nach kurzer Zeit ist das Programm entpackt und startet automatisch. Ganz oben sehen Sie ein stilisiertes 64'er-Logo, darunter einen kurzen Text: »64'er proudly presents: the 64'er intro«, das langsam nach links und rechts scrollt. Daneben findet sich sogar eine vertikale Laufschrift, das Wort »64'er« wandert von unten nach oben. Der Trick ist einfach: Es wird einfach der Zeichensatz so geändert, daß der Eindruck entsteht, als ob das Wort nach oben wandert. Sprites werden für dieses Wort nicht verwendet!

Trickreiche Programmierung

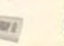
Sprites sind aber dafür zuständig, daß sich die Hintergrundfarbe der Texte in der Mitte des Bildschirmes wellenförmig ändert. Da für die Darstellung reverse Zeichen verwendet werden, überdeckt das, was Sie als Hintergrund der Zeichen wahrnehmen (gesetzte Pixel im Zeichensatz), die wellenförmigen Sprites, die sich hinter dem Text verbergen. Letztere können Sie sich sogar »nackt« ansehen: Steigen Sie mit einem Reset aus und geben folgenden Befehl im Direktmodus ein:

```
SYS 3416
```

Sofort werden die Sprites sichtbar. Das Intro kann nach einem Ausstieg übrigens mit

```
SYS 4096
```

neu gestartet werden. Sie können und sollten also einen Monitor in den Bereich ab $8000 oder ab $c000 laden, und das Intro genau untersuchen. Dabei wird Ihnen die Tabelle 1 hilfreich sein, in der alle wichtigen Routinen verzeichnet sind.



Schrift am laufenden Band

Ganz unten sehen Sie eine Laufschrift. Genauer gesagt sehen Sie momentan nur den Hinweis, daß Sie einen Joystick in Port 1 zur Steuerung verwenden können. Drücken Sie diesen Joystick nach links. Die Schrift erscheint. Mit dem Steuerknüppel läßt sich jetzt die Geschwindkeit in vier Stufen beeinflussen. Drücken Sie nach links, wird der Text schneller. Bewegen Sie den Stick nach rechts, nimmt die Geschwindigkeit ab. Ein Druck nach unten stoppt das Laufband, wenn Sie den Joystick nach oben bewegen, startet es wieder (sehr langsam). Wenn Sie den Feuerknopf drücken, wird eine »Selbstmord-Routine« bei $1a00 aufgerufen. Diese löscht den gesamten Speicher und beendet das Intro. Es muß dann neu geladen werden, wenn Sie es weiter ansehen möchten. Daher sollten Sie, wenn Sie das Intro analysieren möchten, erst einmal aussteigen und

```
POKE 3136,0
```

eingeben, damit der Feuerknopf keine Wirkung mehr hat. Sonst passiert es leicht, daß man etwa die SPACE-Taste berührt und das Intro neu laden muß. Oder Sie verbiegen den Sprung bei $c66 auf eine Reset-Routine:

```
$0c66 JMP fce2
```

Wenn Sie nun <SPACE> oder den Feuerknopf drükken, wird nur ein Reset ausgeführt, ohne daß der Speicher gelöscht wird.

Hinter den Kulissen

Nun wollen wir noch genauer die interne Arbeitsweise untersuchen. Eine wichtige Rolle wird dabei die Speicherstelle $d011 spielen. Diese wird dazu genutzt, Rastersplits, Hires-Grafik und weitere Hintergrundfarben darzustellen. Es handelt sich um das VIC-Register 17 (Adresse 53265),

Kurzinfo: Intro

Programmart: Demo-Programm
Laden: LOAD "64'ER INTRO",8,1
Start: Mit RUN
Besonderheiten: Nach einem <RUN/STOP> <RESTORE> oder Reset läßt sich das Intro mit SYS 4094 wieder starten. Die Laufschrift wird mit einem Joystick in Port 1 gesteuert.
Programmautor: Dirk Swienty

das erste Kontrollregister. Hier kann unter anderem zwischen Grafik und Text umgeschaltet werden, der ECM-Modus (wird weiter unten genau erklärt) läßt sich über ein Bit dieser Zelle aktivieren, und die vertikale Positionierung des Bildschirms erfolgt über dieses Register.

Zunächst eine Begriffserklärung. Was ist der Unterschied zwischen einem »Intro« und einem »Demo«? Es handelt sich grundsätzlich um eine ähnliche Art von Programmen, der Programmierer zeigt, was er kann. Während Demos aber ein reiner Selbstzweck sind und in der Szene üblicherweise auch unabhängig von anderen Programmen weitergegeben werden (können), wird ein Intro vor ein fremdes (gecracktes?) Programm gestellt. Nach Druck etwa auf <SPACE> wird dann das Hauptprogramm, z. B. ein Spiel, nachgeladen und gestartet. Das Intro zeigt nur auf eindrucksvolle Weise, wer das Spiel geknackt hat. In der Regel sind Intros viel kürzer als Demos.

Ein Intro besteht aus mehreren Komponenten, die meistens sogar von verschiedenen Leuten erarbeitet werden:
- Grafik (hier etwa das 64'er-Logo)
- Musik
- Laufschrift (hier ganz unten am Bildschirm)
- Effekte (z. B. die senkrechte Laufschrift oder die interessanten Farbeffekte)

Auf Musik haben wir in diesem Demo aus Platzgründen verzichtet. Wenn Sie wollen, können Sie noch ein mit dem Soundmonitor komponiertes Musikstück dazufügen, dies kann dann irgendwo in der Routine ab $1000 mit

```
xxxx JSR $c000
```

aufgerufen werden.

Als erstes sollte man sich ein Konzept machen, wie das Demo überhaupt aussehen soll. Z. B. wo sich die Grafik und die Laufschrift befinden sollen, welche Effekte verwendet werden. Es gibt unzählige Möglichkeiten dafür. Oft programmieren findige Freaks neue Tools, wie etwa zum Öffnen des Randes, für Rastersplits, Scroll durch Border und vieles mehr. Da sind dem Programmierer fast keine Grenzen gesetzt. Oder wer hätte vor drei Jahren schon Demos mit einer Qualität für möglich gehalten, wie sie heute von sehr vielen Freaks aus der Szene geschrieben werden.

Adresse	Inhalt/Funktion
$0a50	Text 'proudly presents...'
$0a70	vier Farbtabellen
$0b00	aktiviert neuen IRQ
$0b0d	neue IRQ-Routine
$0c2c	Joystickabfrage
$0c66	Sprung nach $1a00 (Selbstmord)
$0cd8	Text 'Joystick in Port 1'
$0d58	Sprites initialisieren
$1000	**Start des Hauptprogrammes**, Intro einschalten
$1100	Waagerechtes Hin- und Herschieben der Schrift »64'er proudly presents« usw. in der Bildschirmmitte
$1114	Blinken bestimmter Buchstaben in der Scroll-Line ganz unten
$1134	Farbeffekte hinter der Scrollzeile (Jalousie) in den Bildschirm schreiben
$114b	senkrechtes Scrolling »64'er« in der Bildschirmmitte
$115f	Farbeffekt hinter der Scroll-Zeile (Jalousie, erstreckt sich über die ganze Bildschirmbreite)
$1a00	Selbstmordroutine, wenn der Feuerknopf gedrückt wird
$1abe	erzeugt die Scroll-Zeile ganz unten:
$1abe	horizontales Softscrolling
$1ad2	Zeile scrollen
$1adf	neues Zeichen an den rechten Rand
$1b00	Textspeicher
$2000	Hires-Grafik (64'er Logo ganz oben), Zeichensatz

Tabelle 1. Übersicht über den internen Aufbau des Intros

Beginnen wir mit der Hires-Grafik. Folgendes muß dabei beachtet werden: Vor dem Einschalten des Grafikmodus müssen noch der Bild- und Farbspeicher initialisiert werden. Nehmen wir als Beispiel eine Grafik des weit verbreiteten Malprogramms »Koala Painter«. Laden wir ein Bild dieses Programmes nach $2000, so liegt die Bitmap von $2000 bis $3f40, der Farbspeicher I von $3f40 bis $4328, und das Farb-RAM von $4328 bis $4710. Also müssen die

Das Wichtigste ist ein gutes Bild

Bereiche ab $3f40 nach $0400 und von $4328 nach $d800 ins Farb-RAM kopiert werden. Die entsprechende Aufbauroutine unseres Demos beginnt bei $1000.

Folgende Befehle werden für den oberen Bildschirmbereich ausgeführt:

```
0b14 lda #1d
0b16 sta $d018
0B19 lda #18
0B1B sta $d016
0B1E lda #3b
0B20 sta $d011
```

Diese Befehle entsprechen den Basic-Befehlen

```
POKE 53272,29
POKE 53270,24
POKE 53265,59
```

und dienen dazu, den Grafikspeicher nach $2000 zu legen, den Zeichensatz an Adresse $3000 beginnen zu lassen, den Multicolor-Modus und die Grafik einzuschalten.

Nun wollen wir uns mit der Programmierung von Rastersplits auseinandersetzen. Jeder von Ihnen, der schon einmal ein Demo gesehen oder sogar programmiert hat, wird wissen, wie ein »Rasterbalken« aussieht: Er wird durch mehrmaliges Einlesen von Farbdaten auf den Bildschirm bzw. in den Bildschirmrahmen dargestellt. In unserem Beispiel finden Sie zwei dieser Balken, die den Bereich eingrenzen, in dem sich die senkrechten Scrolltexte befinden.

Wir erläutern nun, wie es möglich ist, in einer Rasterzeile mehrere Farben darzustellen.

```
0B6D ldx #00        Zähler
0B6F lda $D012      Rasterzeile laden
0B72 and #07        vertikale Position bestimmen
0B74 sta $D011      und fixieren
0B77 lda $0A70,x    Farbwerte aus einer Tabelle
                    lesen und
0B7A sta $D020      auf den Bildschirm und
0B7D lda $0A86,x    in den Rahmen schreiben
0B80 sta $D021
0B83 sta $D021
0B86 lda $0A9c,x
0B89 sta $D020
0B8C sta $D021
0B8F lda $0AB2,x
0B92 sta $D020
0B95 inx            X-Register erhöhen
0B96 cpx #16        und für 22 Farbbalken
0B98 bne OB6F       fortfahren
```

Da der Bildschirmhintergrund breiter ist als der linke oder rechte Bildschirmrahmen, muß der entsprechende Farbwert aus Timing-Gründen zweimal gesetzt werden ($0b80). Mit dieser Routine werden aus den entsprechenden Bereichen Farben eingelesen (22 Farbwerte) und nach $d020 (Bildschirmrahmen) und $d021 (Bildschirm) geladen. Geschickt gewählt ergeben diese Farbkombinationen den Rundungseffekt eines Balkens: Es scheint so, als würde sich der Balken aus dem Bildschirm herauswölben (zumindest mit ein wenig gutem Willen). Durch Verschieben der Farben lassen sich weitere interessante Effekte erzielen.

Die drei Befehle ab $b6f dienen zur Synchronisation des gesamten Intros und sorgen dafür, daß praktisch nichts flimmert. Steigen Sie einmal aus und machen Sie den Befehl ab $b74 unwirksam, indem Sie eingeben:

```
POKE 2932,205
```

(Der normale Inhalt ist 141). Starten Sie jetzt das Intro wieder, wird es Ihnen wohl nicht mehr so gut gefallen: Der gesamte Bildschirm flimmert.

Die Farbwerte, die in den vier Tabellen ab $a70, $a86, $a9c, $ab2 gespeichert sind, liefern wir Ihnen natürlich auch mit. Sie können sie aber frei wählen. In unserem Intro wurden folgende Farbwerte verwendet:

```
0a70 .byte $f,0,6,e,f,6,e,3,f,6,e,3,1,f,6,e,3,1,3,
           e,6,0
0a86 .byte $f,0,2,a,f,2,a,7,f,2,a,7,1,f,2,a,7,1,a,
           2,2,0
0a9c .byte $f,0,9,5,f,9,5,d,f,9,5,d,1,f,9,5,d,1,d,
           5,9,0
0ab2 .byte $f,0,b,c,f,b,c,f,f,b,c,f,1,f,b,c,f,1,f,
           c,b,0
```

Diese Werte sind hexadezimal angegeben, entsprechend den Farbcodes im Handbuch, also 0 = Schwarz, 1 = Weiß, a = 10 = Hellrot, f = 15 = Hellgrau etc.

Neuartige Rasterinterrupts

Sie konnten in Ausgabe 43 des Sonderheftes nachlesen, was Rasterinterrupts sind. Damals haben wir die übliche Methode beschrieben, sie zu erzeugen: IRQ-gesteuert; bei bestimmten Rasterzeilen wird ein IRQ ausgelöst, der dann z. B. die Farbe umschaltet. Auch im 64'er-Intro werden Raster-IRQs verwendet, die hier allerdings auf eine andere Art und Weise erreicht werden. Nehmen wir an, Sie wollen die deutsche Fahne auf dem Bildschirm erzeugen. Dazu gehen Sie bei der neuen Technik wie folgt vor:

```
      sei
   l1 lda $d012
      cmp #1
      bne l1  (warten, bis Rasterstrahl bei
               Zeile 1 ist)
      lda #0
      sta $d020
      sta $d021 (in Zeile 1 Bildschirm auf Schwarz
                 schalten)
   l2 lda $d012
      cmp #$56
      bne l2  (warten, bis Rasterstrahl bei Zeile
               86 ist)
      lda #2
      sta $d020
      sta $d021 (in Zeile 86 Bildschirm auf Rot
                 schalten)
   l3 lda $d012
      cmp #$aa
      bne l3  (warten, bis Rasterstrahl bei
               Zeile 170 ist)
      lda #7
      sta $d020
      sta $d021 (in Zeile 170 Bildschirm auf Gelb
                 schalten)
      jmp l1    (Endlosschleife)
```

Man kann wohl schon ganz gut erkennen, wie diese Methode funktioniert: Der Computer tut eigentlich kaum mehr etwas anderes, als auf den Rasterstrahl zu warten. Ist er in Zeile 1, also ganz oben angekommen, wird der Bildschirm auf Schwarz geschaltet. Jetzt wartet der Computer auf die Bearbeitung der Rasterzeile 85, die etwa am Ende des oberen Drittels liegt. Hier wird von Schwarz auf Rot umgeschaltet. Dasselbe geschieht dann ganz unten mit der gelben Zone. Auf diese Weise wird die deutsche Flagge erzeugt... Nach der gelben Farbzone springt das kleine Programm an den Anfang zurück, um wieder auf Zeile 1 zu warten und die Fahne von neuem zu erzeugen.

Diese Routine kommt in dieser Form im 64'er-Intro nicht vor. Aber im Prinzip arbeitet auch es nach dieser Methode, um ganz oben am Bildschirm Multicolor-Grafik (das 64'er-Logo) darzustellen. Weiter unten kommen die Rastersplits, danach der kurze Text, der nach rechts und links gescrollt wird, dann wieder ein Split, und schließlich ganz unten das Schriftband. Die entsprechenden Routinen, die in der Tat vom Prinzip her wie das Flaggenprogramm aufgebaut sind,

Der Textmarker

finden Sie im Intro ab Speicherzelle $0b0d (neue IRQ-Routine). Die IRQ-Routine läuft hier in einer Endlosschleife.

Beispielsweise wird ganz oben am Bildschirm die Hintergrundfarbe auf Schwarz gestellt. Das läßt sich freilich ändern: Stoppen Sie das Intro, geben Sie folgendes ein:

```
POKE 2852,11:SYS 4096
```

(der normale Inhalt ist 0) und bewundern Sie die neue Farbe.

Die Routine, die das Bild selbst aufbaut, beginnt direkt bei $1000. Wenn Sie nach einem Ausstieg eingeben:

```
POKE 4274,96:SYS 4096
```

(der normale Inhalt ist 76), können Sie das gut erkennen: nur das Bild ohne Effekte erscheint.

Kommen wir nun aber wieder auf die Speicherstelle $D011 zurück und widmen uns dem Modus, in dem man mehr als eine Hintergrundfarbe benutzen kann. Man nennt ihn »Extended Color Mode«, abgekürzt »ECM«. Es können jetzt nur noch 64 Zeichen aus dem Zeichensatz verwendet werden; die beiden höchsten Bits im Bildschirmcode bestimmen die Farbquelle für die Hintergrundfarbe: Zu der üblichen Bildschirmfarbe ($d021) haben wir nun eine weitere ($d024). In der Laufschrift im Demo sehen Sie, daß die Zeichen in drei verschiedenen Farbtönen blinken, als ob man bestimmte Textteile mit einem Textmarker (Leuchtstift) markiert hätte. Dieser Modus wird im Intro ab $0c04 eingeschaltet, indem der Wert $5b in das VIC-Kontrollregister $d011 geschrieben wird. Normal eingegebene Zeichen behalten den Farbwert von $0286 (Cursorfarbe) bzw. die Farbe des entsprechenden Wertes aus dem Farbspeicher. GeSHIFTete Zeichen erhalten den Farbwert, der in $d022 steht. Reverse Zeichen erhalten die Farbe aus $d023 und reverse geSHIFTete Zeichen den Farbwert der Adresse $d024. Um einen Blinkeffekt wie in der Laufschrift in dem Demo zu erreichen, müssen nur diese drei Farbwerte kontinuierlich verändert werden. Diese Routine finden Sie im Demo ab $1114.

Die Laufschrift

Wie das Textband am unteren Bildschirmrand funktioniert, ist schnell erklärt. Es handelt sich um Buchstaben, die im Lores-Modus in die zweitunterste Bildschirmzeile geschrieben werden (nach dem Start des Intros). Soll nun die Bewegung nach links stattfinden, erfolgt zunächst das »Softscrolling«. Das heißt, im VIC-Register $d016 werden die untersten 3 Bit, die die waagerechte Position des Bildschirmfensters in Pixelschritten festlegen, langsam von Null bis sieben durchgezählt. Die Routine befindet sich ab $1abe. Danach erfolgt ein »Hardscrolling«, die 40 Buchstaben der Zeile werden im Bildschirmspeicher um ein Zeichen nach

links geschoben (ab $1ad2). Genau gleichzeitig werden die 3 Bit im VIC wieder auf Null gesetzt, damit die Schrift sich zitterfrei bewegt. An den rechten Rand wird das neue Zeichen des Textes geschrieben ($1adf). So erfolgt je nach Scroll-Geschwindigkeit immer wieder nach einer bestimmten Anzahl IRQs das Softscrolling, und nach jeweils acht Softscroll-Vorgängen führt der C 64 das Hardscrolling durch.

Farbeffekte

Hinsichtlich der Rasterfarben-Programmierung sehen wir uns nun einmal eine Routine an, die durch entsprechende Farbwertverschiebung einen netten Effekt ergibt (im Hintergrund der Laufschrift). In einem Bereich verschieben wir eine beliebige Anzahl von Farbwerten jeweils in einem Kreislauf um einen Wert nach rechts. Der ehemals erste Farbwert wird zum neuen letzten Wert. Im 64'er-Intro funktioniert das wie folgt: Wir verschieben die Inhalte der Speicherzellen $e00 bis $e30 untereinander im Kreis herum und laden den Bereich von $e00 bis $e04 nach $ae4, $ae8 und $aec. Dies sieht nun so aus, als ob eine Jalousie auf und zugeht. In $ae4 bis $aef stehen die Werte, die dann später als Rasterfarben eingelesen werden (die entsprechende Routine findet sich ab Adresse $1134). Das Verschieben selbst findet bei $115f statt.

Einen kleinen Tip wollen wir Ihnen noch mitgeben. Wie bereits erwähnt, können Sie durch Druck auf den Joystick-Feuerknopf das Intro verlassen. Wenn Sie anstelle der Selbstmord-Routine, die bei $1a00 beginnt, von $0c66 aufgerufen wird, den gesamten Intro-Speicher löscht und dann zum Basic-Warmstart springt, eine eigene Routine einbauen, die ein bestimmtes Programm von Diskette nachlädt und startet, haben Sie ein echtes, praktisches und sinnvolles Intro.

Nach diesen vielfältigen Anregungen wird es Ihnen sicher nicht mehr schwerfallen, eigene Demos in noch besserer Qualität zu programmieren. Wenn Sie eigene Werke an uns einschicken, würden wir uns sehr freuen.

(Dirk Swienty/Nikolaus Heusler/ef)

```
Name : 64'er intro          0801 1c07
-----------------------------------
0801 : 0e 08 e8 fd 9e 36 39 33  f4
0809 : 30 20 92 21 00 00 00 78  03
0811 : a2 ff 9a a0 00 84 01 a2  db
0819 : cc bd 57 08 9d 33 03 b9  8e
0821 : 1d 09 99 ff 00 ca d0 02  c7
0829 : a2 01 88 d0 ec c6 2e a5  91
0831 : 2e c9 08 f0 0d ce 3d 08  b0
0839 : b1 2d 99 ff ff 88 d0 f8  61
0841 : f0 eb 84 f8 a0 0a a9 ef  48
0849 : 85 f9 a9 80 a2 8f 86 2d  60
0851 : a2 1b 86 2e 4c 34 03 a2  a0
0859 : 00 86 fe 0a f0 20 90 3c  68
0861 : a2 06 0a f0 2b 26 fe ca  1c
0869 : d0 f8 a6 fe 48 bd 07 01  cf
0871 : 8d 01 08 68 ee 4e 03 d0  9d
0879 : de ee 4f 03 d0 d9 b1 f8  97
0881 : 2a c8 d0 da e6 f9 d0 d6  ce
0889 : a9 37 85 01 58 4c 10 08  88
0891 : b1 f8 2a c8 d0 cf e6 f9  7d
0899 : d0 cb f0 ec 0a f0 15 b0  06
08a1 : 2b a2 03 0a f0 1a 26 fe  96
08a9 : ca d0 f8 a6 fe 48 bd ff  17
08b1 : 00 4c 4d 03 b1 f8 2a c8  a8
08b9 : d0 e5 e6 f9 d0 e1 f0 c8  e6
08c1 : b1 f8 2a c8 d0 e0 e6 f9  36
08c9 : d0 dc f0 bc 0a f0 15 b0  b9
08d1 : 38 a2 06 0a f0 1a 26 fe  94
08d9 : ca d0 f8 a6 fe 48 bd 47  d6
08e1 : 01 4c 4d 03 b1 f8 2a c8  d9
08e9 : d0 e5 e6 f9 d0 e1 f0 98  b6
08f1 : b1 f8 2a c8 d0 e0 e6 f9  66
08f9 : d0 dc f0 8c b1 f8 2a c8  22
0901 : d0 0c e6 f9 d0 08 4c 65  19
0909 : 03 a2 07 0a f0 ee 26 fe  7e
0911 : ca d0 f8 a6 fe 48 bd 87  8e
0919 : 01 4c 4d 03 00 ff 05 b7  78
0921 : 53 01 03 0f 0e e8 20 06  4c
0929 : d0 e0 07 f0 0b 08 3c ef  0b
0931 : e1 a9 0a 09 8d 0c e6 ea  35
0939 : 14 13 a0 c0 04 0d 12 fe  06
0941 : 18 02 19 38 1a 1c 1f a2  ec
0949 : fb 28 2c e7 17 40 b0 10  b7
0951 : 85 bd cf 2e 80 f1 9d 2f  db
0959 : 30 34 fc 4f 99 60 81 15  99
0961 : 2a 70 af c3 2d 66 11 4c  0a
0969 : a1 50 8f df ee fa e2 16  91
0971 : 3f 4a c8 f3 24 b1 c9 eb  55
0979 : fd 4b ab f2 58 77 90 9f  28
0981 : a5 f8 1b 21 7e 92 ad c6  4e
0989 : f5 f6 f9 39 3e 41 7c 83  86
0991 : 88 8c bf ce ec f7 35 3b  03
0999 : 5f 6f 78 86 93 a4 a8 b8  11
09a1 : cc e9 57 7f aa c7 ca dc  f5
09a9 : e3 23 27 2b 36 3a 46 4e  38
09b1 : 51 87 8e 91 a6 ae b4 b9  c2
09b9 : ba bb bc c5 cd e5 1d 1e  f5
09c1 : 22 31 32 37 48 55 56 59  2b
09c9 : 62 65 84 8b 96 9c a7 ac  b7
09d1 : d9 dd 29 33 43 44 5b 5e  ca
09d9 : 64 69 7d 8a a3 b3 c1 c4  0b
09e1 : d4 d5 d7 d8 da db e4 25  1b
09e9 : 26 42 45 47 52 54 5a 5d  57
09f1 : 67 6b 6c 75 82 95 98 b2  74
09f9 : cb d2 d3 d6 de ed 3a 49  e2
0a01 : 4d 5c 61 63 6a 71 74 7a  3a
0a09 : 7b 89 94 97 9a 9b 9e b6  cf
0a11 : be c2 d1 f4 00 00 00 00  43
0a19 : 00 00 00 00 00 00 00 00  1a
0a21 : 00 00 00 00 00 0b 69 bd  9b
0a29 : 76 16 b0 19 34 04 9f 0b  f2
0a31 : 23 23 ab d6 88 a4 9a a3  0b
0a39 : d8 80 9b 83 66 f4 1a c8  b1
0a41 : d5 80 c9 a4 a4 f8 04 94  a8
0a49 : 74 45 8a 21 4e 64 44 a3  87
0a51 : 44 96 c9 82 4a 48 91 d3  78
0a59 : c8 e8 f5 6b f3 88 97 51  05
0a61 : 23 4c 92 c3 11 27 35 9f  26
0a69 : d8 54 c2 0f 5c 94 42 34  da
0a71 : d1 20 b4 d6 d4 48 d3 6c  12
0a79 : 0d 62 a4 6d c7 62 b6 76  e5
0a81 : 47 24 6d ec 4e a3 92 36  8c
0a89 : f6 29 af a7 10 0b db 1b  f4
0a91 : 10 1a ae a8 f7 a3 92 36  c2
0a99 : d8 f4 02 39 bb ed 11 63  c9
0aa1 : 71 38 80 69 7b 6c 40 26  e4
0aa9 : ae d2 df 58 09 14 eb 1f  e2
0ab1 : 6d 1c dd f6 88 b1 b8 9c  b5
0ab9 : b6 d2 f6 5d 27 02 da 82  35
0ac1 : 82 34 c9 a6 9d 90 40 44  8c
0ac9 : ba a3 de a3 40 56 02 45  ca
0ad1 : 00 8e 63 ed ae 6d a0 fb  80
0ad9 : 98 00 00 00 00 00 00 00  72
0ae1 : 00 00 01 6d 7e 6e 36 65  ce
0ae9 : 01 b3 b4 d9 2c ce fd 89  70
0af1 : 49 bd 76 15 47 bd 5e 80  36
0af9 : 93 58 09 14 07 80 01 59  a8
0b01 : 11 3a 9c de 92 8d 11 27  5a
0b09 : 59 ca 64 dc 79 3e 02 05  18
0b11 : 64 45 8e 30 bd 08 30 17  cc
0b19 : 19 52 3d a9 76 89 76 92  93
0b21 : 51 24 38 ec df 20 49 83  5b
0b29 : 10 c8 8d 65 4e 76 12 52  33
0b31 : 23 32 b7 40 28 b5 97 66  bf
0b39 : f9 ac fe c2 a6 10 60 d6  ba
0b41 : 49 e8 d9 31 d6 6b 54 af  14
0b49 : 70 44 b1 34 71 1d 5d 81  47
0b51 : 48 f6 6a f5 95 5a c5 95  dc
0b59 : 72 14 72 b2 3a ec 82 68  ae
0b61 : a4 da 58 a2 35 95 10 c5  a9
0b69 : 96 24 b0 19 a6 68 3a 00  f7
0b71 : 00 05 ba 56 e9 52 00 01  a0
0b79 : 6e 97 6a 56 c0 00 5b 8b  a9
0b81 : 16 2d b0 00 1d a9 52 a5  0e
0b89 : d8 00 05 b8 a9 d8 a4 86  bb
0b91 : 39 38 82 9d 8b 14 00 02  98
0b99 : 82 2d ba 4a 00 00 14 89  0d
0ba1 : 19 ab 92 97 c4 79 57 2e  f9
0ba9 : 5c 94 00 02 04 08 0a f5  24
0bb1 : 50 00 0a 4b 04 ea c2 90  b1
0bb9 : 00 08 b1 62 90 4a f9 29  0c
0bc1 : 5a 56 e9 10 54 a4 f3 a4  46
0bc9 : a5 b6 88 24 3f d2 7d 71  d3
0bd1 : a5 48 90 d5 d2 43 cf b4  69
0bd9 : 58 a4 16 a9 28 7a 48 82  bb
0be1 : 43 57 49 dc 1e d4 a9 10  0d
0be9 : 51 24 fe b7 02 d8 00 05  f4
0bf1 : 05 cb 97 2e 10 51 64 fe  a3
0bf9 : a5 4a d9 05 16 48 69 2a  78
0c01 : af 90 51 64 a3 29 5b a4  13
0c09 : 41 34 13 c2 aa a5 48 82  7f
0c11 : 42 cf 43 a8 21 90 4c 0c  01
0c19 : 90 c0 bf 16 d8 25 0e c7  3a
0c21 : 49 40 a3 6b a2 ab a3 49  89
0c29 : d8 12 87 62 8c a0 02 e4  d8
0c31 : a2 43 6c 27 fe 5a eb e8  b9
0c39 : bf 5f f4 d6 39 20 24 36  51
0c41 : c6 78 79 dc 02 05 02 81  91
0c49 : 1d 80 20 50 21 40 85 02  e7
0c51 : 81 05 c5 47 99 50 d5 57  d1
0c59 : 00 d2 20 68 dc e8 81 62  b7
0c61 : 81 29 03 2a fb 98 81 46  94
0c69 : 02 1d 1b 83 b0 0a 37 3a  de
0c71 : 74 e9 72 88 3a 5c a3 46  29
0c79 : 8d ce 80 02 06 50 65 20  86
0c81 : 40 00 5c b8 a0 b9 70 70  6a
0c89 : c1 72 e7 40 00 0a 02 09  f0
0c91 : a9 9e 10 00 d0 e8 dc e9  29
0c99 : e4 00 b6 48 c9 59 e1 c9  b6
0ca1 : 5b 0c a9 58 70 e1 00 15  b8
0ca9 : 4a 3c 3f 62 db 00 2d a8  f1
0cb1 : e8 c3 a5 6c 00 57 ab ed  b7
0cb9 : c0 81 00 00 b7 17 b4 3a  b5
0cc1 : 56 c8 25 3c f9 04 13 35  c3
0cc9 : 3c 36 e9 51 20 9c 69 f2  37
```

```
0cd1 : 2d d2 55 52 a5 6c 82 9b  06
0cd9 : 27 54 35 1d b0 00 04 16  63
0ce1 : 39 ec 38 00 0b 80 05 cb  ff
0ce9 : 9d 20 5c e9 e7 a5 c8 00  b9
0cf1 : 00 14 00 50 0e d0 1b 11  fb
0cf9 : c9 24 26 02 4c 74 9f c4  0f
0d01 : b2 fc 3f bf 22 9b 73 a8  17
0d09 : 43 3a b4 bc 5d fb a2 56  1b
0d11 : a8 6a 73 bf 23 38 84 8c  e2
0d19 : 36 39 28 74 b0 4a dd 31  bc
0d21 : 91 d8 2f 99 22 b0 fc 32  1d
0d29 : 22 53 c1 61 f8 05 2f cd  a1
0d31 : d0 c9 dc f3 92 21 43 1d  15
0d39 : 1a 86 0f 7d 76 8f ee c5  35
0d41 : 1b d0 b5 da 15 51 40 a9  bd
0d49 : 56 a4 91 42 62 3a b0 c2  de
0d51 : b5 0a 7c 0a 96 8f 08 e0  33
0d59 : 87 00 f0 e0 18 f0 e0 18  f5
0d61 : a7 87 00 c5 34 08 60 38  fa
0d69 : 2e ce 3d d9 d1 8f 76 74  e5
0d71 : 62 9e ec e8 c5 8d 3a e8  fe
0d79 : 0e 09 e2 3c f1 59 3c f1  0b
0d81 : 59 29 e7 8a c9 6c 8a 78  d5
0d89 : 0e 09 14 4f 22 89 cf 22  fd
0d91 : 89 ca 79 14 4e 53 d1 90  48
0d99 : 09 12 41 20 62 92 51 8a  15
0da1 : 4c f9 32 64 c9 9f 26 4c  ce
0da9 : 38 06 1d 10 1a 04 31 c8  46
0db1 : 02 db 1d 9a d2 18 ea 86  e2
0db9 : 56 fb 32 63 cc d6 21 98  3f
0dc1 : b4 f7 44 c9 35 9a a1 72  4f
0dc9 : 24 db 94 11 71 26 ac e8  ef
0dd1 : 20 a2 4d 05 08 51 28 27  30
0dd9 : e2 4d 69 d9 8b 51 22 64  8c
0de1 : 9b 56 85 c8 93 64 d0 45  4c
0de9 : c4 9a 9e a0 82 89 8c 0c  75
0df1 : b2 3a 72 b2 22 53 6b 05  28
0df9 : 38 90 d7 f2 02 78 31 6e  53
0e01 : c4 58 4d 63 88 8b 63 64  ec
0e09 : 21 95 3e 92 29 b0 bb c5  69
0e11 : 54 e2 fc ca 64 8e f4 c5  89
0e19 : c8 87 51 5a 88 d7 46 a0  e6
0e21 : 82 89 b7 ec ea 10 a2 6d  88
0e29 : 5b ce a0 9f 89 41 3f 13  cd
0e31 : 6b 52 48 c9 c9 12 31 07  11
0e39 : 6b a7 3a 84 28 90 62 a2  6d
0e41 : e2 37 b3 52 cc c5 a8 84  9d
0e49 : 8c e7 48 43 4b 44 d3 9b  a1
0e51 : 3c 82 57 48 96 4a 2f 1c  5e
0e59 : 87 4d 2e 89 9c 9f e2 8a  ab
0e61 : 43 33 04 9c dc c8 c4 89  0d
0e69 : fc f6 86 5d 32 7b db 20  dc
0e71 : 3d cb c4 e9 b4 68 4f 2e  2a
0e79 : 3b 2b 2d 3d 1d 87 d2 0a  aa
0e81 : b4 21 44 da 8b ab 41 3e  ca
0e89 : 41 33 32 32 66 aa 48 cc  ad
0e91 : 24 89 69 53 b8 b5 49 3d  17
0e99 : 61 20 28 64 32 24 34 c9  4a
0ea1 : 3b dd 15 a7 30 e2 1d 85  9f
0ea9 : 12 f3 05 04 5c 49 ad 95  69
0eb1 : 04 14 49 b2 29 d3 a7 0b  4e
0eb9 : aa 0a 69 48 26 8e 88 61  87
0ec1 : 01 05 0a a2 0a 13 d9 82  c1
0ec9 : 13 6b 45 25 6e 3d 54 92  cf
0ed1 : c6 3e 84 92 52 3a 79 25  51
0ed9 : 92 3e 24 94 b8 ec 2a 0f  e0
0ee1 : d5 39 a5 a1 ff 02 0f d5  e8
0ee9 : 39 b7 68 7f c0 83 f5 4e  a4
0ef1 : 68 28 7f c0 83 f5 4e 6c  5f
0ef9 : 92 43 c7 45 12 14 18 30  4a
0f01 : 42 4a a0 a1 2d 10 4a 22  85
0f09 : b0 7e 46 da 02 71 ef 55  fb
0f11 : a9 3c d2 21 18 f0 2a 5a  17
0f19 : 3d 1a 22 85 ac f6 aa 42  4e
0f21 : 61 59 d1 c6 01 47 38 01  a9
0f29 : 08 fb 82 68 68 7b 89 43  ec
0f31 : 0c 49 bc 28 6b 89 8c 13  71
0f39 : 55 1c 93 b1 a6 9b 93 0d  67
0f41 : 12 0c 58 71 38 cd 76 a8  bb
0f49 : 5b 8d 40 31 a6 9a 82 02  ee
0f51 : 25 0d f1 26 d7 a1 a2 24  9b
0f59 : d4 65 06 6a 24 d8 14 1c  40
0f61 : 38 93 61 d0 71 22 4d 0d  4d
0f69 : 04 f4 4c 60 da a2 b3 c8  2a
0f71 : 11 20 c5 d7 11 49 e7 ea  cf
0f79 : e6 b2 e6 22 e7 eb 25 59  db
0f81 : 92 88 c9 34 6d c6 b6 e0  fa
0f89 : 6e e0 2e dd 9d 3a 34 62  f0
0f91 : 94 c6 81 02 1c 31 af 40  b4
0f99 : 55 8b b1 a4 0a 94 a3 76  75
0fa1 : c6 2a 6c 31 85 7b db 31  c4
0fa9 : a9 c0 c4 7d 05 96 a4 16  57
0fb1 : 5a 90 31 1f 40 d6 57 03  a2
0fb9 : 59 5c 9a f8 59 44 a5 db  0c
0fc1 : 74 b8 78 85 13 91 12 a5  b1
0fc9 : 0b 59 5c 94 78 59 75 14  7b
0fd1 : a8 cd d9 23 b9 b5 c7 78  94
0fd9 : 12 8d 5d be b3 f5 b5 52  47
0fe1 : a6 d8 07 29 38 d4 4d 78  2b
0fe9 : 1b 66 a9 c0 fe ee ac 81  d7
0ff1 : 1c e3 72 00 b6 9a 7a 0b  dc
0ff9 : 7b b0 9b 16 67 18 36 81  89
1001 : af e4 0b 1b 57 7d 79 01  92
1009 : bf 83 14 11 25 e3 06 d0  dc
1011 : 75 e4 01 6d 5d ac b9 01  0a
1019 : ac 83 15 76 41 41 b3 93  af
1021 : 2c ee 40 93 06 2e 58 8a  8f
1029 : 4c 60 da 85 9d c9 df 87  c3
1031 : b7 cc ee 4d cc 38 31 75  f2
1039 : c4 97 34 14 34 9a 12 5f  78
1041 : 04 d8 d4 17 98 31 82 68  b7
1049 : 39 26 1c 08 31 [illegible] d1 14  ab
1051 : b0 93 16 48 7a 96 42 e6  8d
1059 : 85 6b ae 42 a6 5c b0 a6  e5
1061 : c9 0d 29 47 d6 99 ba 68  da
1069 : 79 87 41 33 0e 6a a6 82  30
1071 : b2 1d 04 dc 39 ae 2d 0d  26
1079 : 50 e8 30 30 e6 b6 14 1b  fa
1081 : f8 74 25 c3 c6 0d ad fd  fd
1089 : 1e 4f 54 68 31 70 44 52  bd
1091 : 12 1e aa 21 63 01 20 68  11
1099 : d0 32 3a 4d b4 16 a8 2e  b6
10a1 : a4 41 8b 86 25 dc 60 63  1b
10a9 : b4 9b cb 22 a9 36 dc 36  8e
10b1 : a8 33 91 36 b1 36 a8 30  ee
10b9 : b1 36 a6 2d 50 48 44 83  34
10c1 : 15 9b 20 94 30 3d 2a 4d  6f
10c9 : e7 18 19 67 50 82 4d c8  cf
10d1 : e7 30 93 d3 92 3b 83 f7  31
10d9 : 16 03 dc 45 87 96 ac 0c  49
10e1 : f6 30 6d 2b 3d c8 0c f4  e4
10e9 : 18 a9 88 8a 45 6a 61 9e  b4
10f1 : f3 8c 1b 4b 03 93 85 3b  b4
10f9 : 94 19 dc 96 13 a0 c5 14  99
1101 : 49 6d 5f 48 18 fa 6d 04  f9
1109 : 0e 45 80 c1 47 32 0d 32  b1
1111 : 82 56 96 07 91 ad 81 26  1e
1119 : 4c f9 f3 a7 48 91 46 8d  99
1121 : 9b 30 20 1c f0 20 59 b3  3d
1129 : 46 8c 89 13 a7 4f 9f 0e  0a
1131 : 1c 08 06 31 27 b6 96 71  5e
1139 : f7 fc 86 99 88 93 dc bb  93
1141 : 21 9c 8b 0a 8e 34 69 d3  ac
1149 : ae dd 1c 60 48 91 46 89  36
1151 : ce 52 9c f4 68 c8 90 36  89
1159 : 3c 04 8d c6 72 df 24 b3  f2
1161 : 94 d0 21 9c d0 21 9c 82  c7
1169 : 52 ce 90 48 d1 a4 47 c9  42
1171 : 1c e5 8d 3a e9 e3 4e ba  97
1179 : 72 0a 8c 84 a9 23 33 52  c9
1181 : 10 e1 57 1c b6 45 3c f6  50
1189 : 45 3c e4 13 39 21 0e 48  ed
1191 : cd 7c 80 f3 e6 41 2c a4  ad
1199 : 6e 39 e8 c8 3d 19 00 38  04
11a1 : c2 01 a5 80 51 d8 06 f1  35
11a9 : 84 03 08 06 10 0c 20 1a  88
11b1 : 90 0b 68 5e d2 90 9b 84  d6
11b9 : a4 26 c8 33 40 48 6c bc  7a
11c1 : 0e 3c d3 18 c1 b5 01 83  ba
11c9 : 94 a3 41 8a b2 88 a4 c7  62
11d1 : 8a 68 39 01 bc 60 b5 a7  f3
11d9 : 50 72 8c b6 9b d7 61 71  3d
11e1 : 83 6f d6 23 91 48 26 a2  6f
11e9 : 4a d1 a0 32 21 c6 34 47  32
11f1 : 66 68 28 02 4c d9 38 d2  f0
11f9 : 9b 37 ec fc b7 40 2d d9  f0
1201 : 12 09 ce 93 de 13 46 c8  ef
1209 : 20 d9 16 a9 31 09 a3 64  83
1211 : 10 6c 8b 54 36 84 0a 25  bf
1219 : e6 0b eb d9 37 28 2f 30  8d
1221 : 74 37 90 46 24 67 0e 06  df
1229 : e0 87 6f 88 dc 9d 28 d0  b7
1231 : 62 b0 d1 14 8a 26 81 3d  3d
1239 : 1e 05 05 74 64 44 5c 11  a6
1241 : 0c 88 99 81 06 c1 60 9a  4d
1249 : 0a 11 70 4d 4f 50 99 81  82
1251 : 06 87 c0 0d fb e1 51 9e  3e
1259 : 2b a2 b5 0a 88 aa 8a d4  36
1261 : 2b 02 ab 09 eb 64 9a ad  41
1269 : 8a 1a cb 84 33 1c d3 6c  c0
1271 : 50 ab 36 2d de 6d 8b 8c  6b
1279 : d0 b4 4f b5 0a ac f8 16  44
1281 : 21 55 14 9b 50 aa d4 3d  ed
1289 : 1c 10 ac 16 d1 7d c3 cd  4f
1291 : 0a 7c 98 02 81 08 53 e0  a7
1299 : 42 11 a4 d4 15 58 4e db  2c
12a1 : 16 e4 b9 1d e1 6b b4 52  2c
12a9 : 09 98 9c 59 e2 b4 28 10  e5
12b1 : b0 7b 85 20 e2 81 00 50  5f
12b9 : 37 92 1b 07 38 eb 4a b0  4f
12c1 : b4 4f 14 2b 13 c5 54 55  e3
12c9 : 61 68 15 ab 02 81 08 be  e3
12d1 : cd 26 13 ac 92 65 e1 55  92
12d9 : 34 93 c0 84 b8 d8 ac 50  3d
12e1 : d6 74 3d 2b 5a 68 55 45  6f
12e9 : 46 ed 5a 90 b5 da 3c 6b  c8
12f1 : 45 b3 64 82 8b ad 0e 49  6a
12f9 : 30 75 85 3e 14 6a 7e 15  c6
1301 : 59 e2 24 93 20 bf 23 03  d9
1309 : 32 a8 a4 8a 03 13 08 55  9d
1311 : 61 4f ab 50 90 e7 20 b9  4b
1319 : f7 8c 5b 50 a0 4f 34 9a  c2
1321 : 90 aa 9a 4e 81 5a df 0b  f7
1329 : 7e fd f0 8b 02 15 89 e4  0c
1331 : 12 b3 5a 06 ad 41 51 a2  e4
1339 : 28 52 0f 66 c8 a0 42 9f  f5
1341 : 02 12 2e 78 a8 8a 01 24  12
1349 : f1 d7 b8 31 d2 76 27 8a  0d
1351 : 13 1a a1 74 76 2e 17 1b  d4
1359 : 33 42 b1 3c 44 12 e7 59  c8
1361 : fd c0 ab 08 b7 45 6a 15  24
1369 : 53 49 14 09 02 81 be 13  d4
1371 : ce 92 50 c5 63 5d 57 45  5e
1379 : 97 5c 3a db 15 f5 9b a5  03
1381 : 6a f3 64 55 6b 2e b8 65  7e
1389 : d4 31 50 90 a7 c9 b5 0a  d0
1391 : a0 8b 66 15 81 5a 85 54  dd
1399 : f0 2c 0a d4 91 43 a8 2d  ed
13a1 : f0 a1 65 81 be 48 bd d5  bc
13a9 : a2 2b 4d 74 2d fb c9 0f  bb
```

Listing 1. Das 64'er-Intro in gepackter Form. Bitte mit dem MSE (Seite 159) eingeben.

```
13b1 : ea 0b 26 ad e0 18 61 0c   cc
13b9 : 30 86 d2 42 94 38 70 89   09
13c1 : 19 81 3a 5a 3b 82 db 00   ac
13c9 : ca 81 88 1b 42 31 02 dc   49
13d1 : b1 88 08 86 00 62 00 42   31
13d9 : 38 60 8a 3b 20 96 96 18   8d
13e1 : 80 93 a6 31 01 74 ba 18   c9
13e9 : 80 c4 ef 18 81 4d 95 24   eb
13f1 : 35 3d ad 29 51 03 72 bc   c6
13f9 : 62 05 3a 63 10 23 c7 1a   46
1401 : ec 05 28 85 66 a6 14 b5   82
1409 : 6e 03 c2 03 0e 18 13 bc   71
1411 : e0 5d 35 d1 d8 05 7e 1c   0f
1419 : 35 45 8f 16 3e ad 51 62   f3
1421 : d6 1c 60 5f cd 9a fe 6b   8e
1429 : f9 65 4a 95 2b 34 ac b2   87
1431 : af 8e 30 07 62 e9 8d 75   ab
1439 : 0c c7 31 a8 6a 0c 6d 7a   3c
1441 : e0 00 00 e1 2e 52 0c 34   6c
1449 : 5a d1 3e a1 63 fd ae 31   93
1451 : 82 fd f8 f1 f5 47 8f 14   4e
1459 : 76 41 76 78 ec 01 26 ca   21
1461 : 57 1d 80 2e d2 d2 07 0c   25
1469 : 15 c7 18 1a ab 24 2f e5   10
1471 : 48 5f eb 5a b5 6f 35 93   82
1479 : 49 af 23 8c 0b f2 a5 66   a0
1481 : 95 2a 57 5b e3 b1 2b 0c   fd
1489 : a1 c6 01 ae 98 c6 ba 62   13
1491 : ab d1 a8 d6 09 1a c5 d7   52
1499 : d2 bd 6e 6c 9d 61 a8 00   fa
14a1 : af 5c 75 00 71 c3 03 00   1d
14a9 : 38 60 ac 34 58 11 83 86   ec
14b1 : 00 8e 18 02 38 04 17 4a   d3
14b9 : a8 5c 83 5f 02 3e a8 f1   f5
14c1 : 63 d6 d5 14 71 80 75 40   f9
14c9 : ec 4a be 41 53 a2 d3 c3   d3
14d1 : 05 7f a8 74 c8 c4 0a 69   fc
14d9 : c0 00 00 95 aa 29 04 cb   e8
14e1 : 5a ac 9a 42 a1 50 be 7c   11
14e9 : a6 ab 0e 30 25 4a cd 7e   c7
14f1 : 56 5c d7 e5 5f cd 7e 56   33
14f9 : 6c b7 c7 18 17 4c 61 d8   41
1501 : bb 5e bf 0a f5 f8 70 e0   c7
1509 : 39 c0 31 a4 1e 01 95 12   e8
1511 : f5 57 00 01 2d 96 18 88   cb
1519 : 2d 35 52 79 15 03 10 2b   a5
1521 : b2 d9 20 18 62 05 7a fa   f9
1529 : a3 d6 46 66 cd 7f 2d fc   1d
1531 : b9 82 38 60 97 02 74 88   b2
1539 : 12 27 40 01 a7 2a 2f c7   27
1541 : fb 5e 9a ea b8 ec 02 9c   a3
1549 : 09 d3 a4 4e 81 00 6b e0   b6
1551 : 47 8a 31 47 1c 60 4a de   be
1559 : 8c bf 0b a0 01 4d 94 18   99
1561 : 80 89 96 31 01 26 ca 5d   99
1569 : 18 81 15 74 b3 80 6b a6   50
1571 : 24 37 96 d7 14 ac 5a c9   75
1579 : a4 d7 9a c8 54 d0 e3 03   2a
1581 : 2e 5c b9 6e 65 9f 0e fe   a3
1589 : 5c b9 b2 e5 cb 98 71 81   75
1591 : 76 ed d2 0a a5 5a fa 78   fe
1599 : 70 e1 a7 4e 9e 03 8c 0e   fe
15a1 : b5 30 f5 42 a1 7e a1 47   57
15a9 : 00 06 18 80 bd f0 48 cb   df
15b1 : db 59 21 34 f7 d2 12 a5   b1
15b9 : 1e 31 ec ca c3 87 0d 4c   49
15c1 : d5 12 03 8c 08 12 28 8e   c1
15c9 : a0 20 48 87 3a 74 69 f2   4f
15d1 : 06 be 04 5d 5a a2 ea ad   a5
15d9 : 5a b0 d1 60 4c 8e 18 1e   e2
15e1 : 21 9c 15 ab 56 ad 97 2e   99
15e9 : 6c b1 e2 c7 fb 1f 0e 1e   6c
15f1 : a4 8c d2 55 d2 b4 e7 05   b7
15f9 : 76 3b 20 35 d1 88 01 44   a9
```

```
1601 : 0c 40 58 27 65 66 95 d6   b6
1609 : a6 1a da ab 45 d5 1c 90   7d
1611 : d7 d5 56 71 4f 7e fd f5   63
1619 : 99 04 d0 da 75 d9 40 45   f5
1621 : 8f 2b 0f 55 53 06 39 20   3f
1629 : 39 04 e1 55 8b 3a 45 4a   bc
1631 : 95 2a 61 c3 53 30 d1 60   6b
1639 : b2 31 01 32 ec 1e 74 08   ac
1641 : 12 20 22 69 8d 7c 08 b5   61
1649 : ab 6a d5 16 3c 71 c6 01   ce
1651 : 8f 03 34 ac d9 72 b5 a3   54
1659 : 86 07 90 c4 8c bc aa 1e   b5
1661 : 79 35 45 20 97 d5 43 96   2d
1669 : fe 62 0a c5 50 df 39 ce   5a
1671 : 62 09 79 56 b4 74 43 12   a1
1679 : 38 60 5f 90 4b aa 9f a0   95
1681 : 10 4a 0b 41 97 7f 2e 64   98
1689 : 85 44 86 66 d2 3c 78 f1   74
1691 : 63 c7 d5 14 76 40 cd e2   36
1699 : 18 80 c0 4b 18 80 14 48   f1
16a1 : c4 04 42 ba 01 ec 9e 44   ca
16a9 : 86 cd c1 a9 86 56 1a 98   70
16b1 : 70 e1 1c f0 18 c7 b3 1a   fa
16b9 : ca c9 83 8e 41 32 b6 9c   d4
16c1 : 4d 23 40 82 2d 6a d5 b5   e9
16c9 : 56 d5 14 82 05 56 2a 04   13
16d1 : 8b a3 86 06 b7 68 00 00   4f
16d9 : 14 a9 96 0c 00 8b 17 85   ad
16e1 : 71 c8 05 f9 52 a5 4a eb   8a
16e9 : 2b 08 e3 00 c4 12 36 aa   1c
16f1 : 96 18 1b cc 35 2a 54 a9   3d
16f9 : 72 a0 e3 00 90 fd d5 0e   21
1701 : 2b a6 0d 7e 15 f8 70 e1   31
1709 : c2 08 e1 82 2c 78 f1 e2   ac
1711 : 47 fb 1c 71 81 53 0d 91   95
1719 : 10 4b df 61 5a 80 00 02   a1
1721 : 31 89 ed 04 00 00 1c 2a   bf
1729 : e1 88 0d eb 9f 0c 42 87   82
1731 : 2b ad 4c 32 ba e1 a8 38   5a
1739 : c0 10 ac d9 8d 64 f1 86   38
1741 : b5 04 7c d9 6f e5 81 3a   f3
1749 : 78 f3 00 a5 2f 54 2c 78   a7
1751 : 9f 74 e9 d2 3d 00 a9 2a   ce
1759 : a7 59 58 65 10 5a ea f7   df
1761 : 29 8c 63 14 d5 eb d7 d3   ef
1769 : 5e bd 7a e3 8c 09 59 af   97
1771 : ca 3c 6b 26 20 9e 1a 81   5b
1779 : 66 c0 69 52 a5 61 8f 13   ae
1781 : ee 9e b5 30 d4 a8 90 48   97
1789 : 5c 1c 60 1e 01 e7 48 81   43
1791 : 22 00 d6 a0 24 36 8b de   5d
1799 : 48 48 56 b5 9f 28 c4 06   ac
17a1 : 85 d7 18 80 99 72 c6 20   b0
17a9 : 21 d2 83 b0 02 b7 ca 40   b4
17b1 : 00 02 f6 de 21 46 c2 82   a0
17b9 : 54 ae a8 c4 60 c4 06 f9   5f
17c1 : cf 00 00 53 15 a1 69 40   7f
17c9 : 00 69 d3 a6 be 91 ce 20   3b
17d1 : a0 fb c9 1c 28 c3 0c 40   b6
17d9 : ed 92 01 05 7a a4 21 d8   f3
17e1 : 03 3e d6 c7 00 00 26 f5   36
17e9 : c9 ae 02 09 23 ef 24 30   4e
17f1 : 75 c3 29 51 cc 62 05 b6   1e
17f9 : 54 00 04 03 9c 21 24 6c   eb
1801 : 56 a3 c8 43 b6 e0 08 4b   ec
1809 : 78 b7 af 04 21 2d e3 5e   91
1811 : 28 42 12 e2 c5 88 a1 08   72
1819 : 4d 7a f5 e3 04 21 2e 28   ef
1821 : d8 89 0d 8b d1 8b bc 6c   b8
1829 : 58 82 10 fa c4 5b de 82   cd
1831 : 10 de 24 66 c3 40 79 87   b9
1839 : fe 7c f9 ff 84 21 e3 c7   64
1841 : 8b 3a ab 08 43 7a ac 6a   65
1849 : b7 82 10 e2 c5 88 82 69   1f
```

```
1851 : 74 7e bc 5b c4 13 6d a1   7d
1859 : a8 bd 3c a8 22 43 62 b4   33
1861 : 31 4c 85 eb c4 86 d1 68   30
1869 : a5 2c d8 b1 10 4e 3e 80   fe
1871 : de 41 12 1b 45 a3 bf 33   af
1879 : 5e bc 41 39 9a 10 25 e2   31
1881 : 50 84 3e b3 e7 cf 9c 82   8e
1889 : 4c e8 52 de bc 52 09 33   a3
1891 : a0 37 95 6c 09 04 99 d0   78
1899 : 7b c5 bc 41 21 f4 77 55   90
18a1 : de bc 41 67 d0 84 f4 a6   6d
18a9 : 20 9b 3d 0c 7f 16 07 11   4f
18b1 : 43 fc 76 37 7f 0a 3d d9   e8
18b9 : dc ab dd c8 c1 fe 3b 08   0c
18c1 : ff e1 0e 7f e4 86 e0 68   fb
18c9 : 2f 19 7b b8 70 dd 2c 72   06
18d1 : 42 48 6e 0e 84 da 1f 38   a1
18d9 : 7c 78 f0 3b 01 f1 e0 1e   94
18e1 : 01 e3 c0 73 b7 49 bf ea   0d
18e9 : 65 5e 70 f2 f0 8f cf bb   3a
18f1 : c3 26 1f fe 3f b0 2e 97   d0
18f9 : 8a e1 53 23 f3 8e c0 51   06
1901 : f9 f7 6e dd 9d 1e 1d d9   40
1909 : d1 e8 f4 7e 7d c1 0f 8f   9c
1911 : e0 ff e0 21 0e 7c fe b3   75
1919 : e7 1c 30 e7 cf b8 21 0f   7d
1921 : a2 09 af b8 10 87 b2 99   86
1929 : 1f 9f 72 6c 21 29 23 22   6e
1931 : b4 1f f7 16 13 ff f6 11   65
1939 : 84 61 02 15 5a 9d 84 43   bc
1941 : 62 28 42 55 3a 3d 4d 78   a6
1949 : a1 0b 3a ce 97 8f 1e 21   89
1951 : 09 71 19 4d 78 a4 14 4d   9a
1959 : 18 88 26 4b 44 b2 de 47   8c
1961 : 90 4c 1e 83 96 f2 ae f5   b7
1969 : e2 90 5a 74 05 4c a7 28   5a
1971 : 42 12 09 4f 73 50 42 1c   e4
1979 : e1 0e 7c fb b8 e7 dc 9b   75
1981 : dd 9f 88 42 1f 41 f4 34   d0
1989 : 58 7d 01 20 90 b0 ad 0d   44
1991 : 7a fb 8c 6d c9 4e d6 d5   f0
1999 : 65 6b 93 ec 2a b8 25 2a   88
19a1 : 3d 4d c4 77 c2 36 40 14   ac
19a9 : e9 a1 26 4c 99 32 9a 12   30
19b1 : 64 c4 24 c9 88 44 24 c1   78
19b9 : a2 e0 d3 83 06 9d 38 30   bf
19c1 : 69 e0 99 32 9c b9 9c d0   f3
19c9 : 88 49 73 20 cb 98 34 14   51
19d1 : 1f 90 71 d3 a7 4d 09 83   1f
19d9 : e4 1a 7f 25 d3 a7 06 65   ac
19e1 : 39 93 10 9f 10 93 26 4c   ab
19e9 : a7 4f e0 e7 41 45 41 f9   a4
19f1 : f2 9d 3a 73 31 d3 f8 8a   59
19f9 : 83 83 e4 14 27 38 28 44   57
1a01 : ba 5f e4 c8 33 26 41 c6   b4
1a09 : 3b 10 51 43 b1 06 d8 ec   92
1a11 : 41 a7 33 04 1c 78 f1 73   a7
1a19 : f9 8f e2 29 09 83 e4 1f   36
1a21 : 93 25 c1 97 33 04 b8 32   45
1a29 : f0 53 45 0e c4 19 90 66   fa
1a31 : 41 99 31 15 32 9d 39 72   08
1a39 : e5 cb 44 e0 f8 8a 99 83   82
1a41 : e4 1c 74 a0 fe a5 33 9a   84
1a49 : 2a 0c c8 30 70 0e c1 04   38
1a51 : 8e bc 96 a5 32 59 04 d4   3f
1a59 : 5e d3 06 66 09 98 d1 53   32
1a61 : 11 54 d2 f4 e9 4b a7 ce   25
1a69 : 0a 2a 9c 18 30 66 4c 1d   54
1a71 : 88 33 07 62 0c b8 33 20   35
1a79 : cc 97 06 64 b9 78 20 c1   82
1a81 : 83 2f 15 3a 53 26 4b 98   ed
1a89 : a5 1d 18 34 a9 52 a5 32   72
1a91 : 0c b9 90 66 52 99 06 0c   8d
1a99 : c2 0b 2e 0c 89 79 93 26   ed
```

```
1aa1 : 4b 99 48 7a e0 cb 1c 08  07
1aa9 : 38 06 28 32 e0 cc 19 e0  cf
1ab1 : d2 97 2e 5c b8 30 60 c1  78
1ab9 : 97 2e 94 18 32 e5 cb 96  3e
1ac1 : 41 48 c1 66 58 e3 41 50  ae
1ac9 : 87 f2 19 68 1d c1 09 f1  05
1ad1 : 62 08 57 e1 43 e4 03 0f  cf
1ad9 : 15 67 dd 3d 1d 27 5d 27  90
1ae1 : c1 67 1f 24 2c 92 d2 d1  e8
1ae9 : d2 b1 d9 56 71 26 89 67  12
1af1 : 37 25 50 a8 63 85 0f e9  56
1af9 : 3e b4 e3 dc 25 dd 31 8c  45
1b01 : c4 84 c8 9a c9 a5 b2 d3  c9
1b09 : 20 19 35 a5 bb a5 63 a4  78
1b11 : c0 a9 12 a0 1b 85 2d 84  da
1b19 : 2e a9 24 a0 1b 20 1e ab  bb
1b21 : 4c 10 08 93 54 48 45 20  c7
1b29 : 36 34 27 45 52 20 49 4e  d4
1b31 : 54 52 4f 92 92 92 92 92  02
1b39 : 92 92 92 92 92 20 43 4f  e1
1b41 : 4d 49 4e 47 20 55 50 0d  b7
1b49 : 11 50 52 4f 47 52 41 4d  a7
1b51 : 4d 45 44 20 41 4e 44 20  2e
1b59 : 44 45 53 49 47 4e 45 44  c2
1b61 : 0d 42 59 a0 44 49 52 4b  68
1b69 : 20 53 57 49 45 4e 54 59  fd
1b71 : 92 92 92 92 92 92 92 92  70
1b79 : 92 92 92 92 92 2c 20 31  b9
1b81 : 39 38 39 0d 0d 54 45 53  f6
1b89 : 54 45 44 2c 20 52 45 56  6d
1b91 : 49 53 45 44 2c 20 4c 49  e5
1b99 : 4e 4b 45 44 20 41 4e 44  35
1ba1 : 20 50 41 43 4b 45 44 92  b7
1ba9 : 0d 42 59 20 4e 49 4b 4f  2d
1bb1 : 4c 41 55 53 20 48 45 55  62
1bb9 : 53 4c 45 52 2c a0 32 37  cd
1bc1 : 30 37 38 39 0d 0d 28 43  22
1bc9 : 29 4f 50 59 52 49 47 48  f6
1bd1 : 54 20 4d 41 52 4b 54 20  c2
1bd9 : 26 20 54 45 43 48 4e 49  0f
1be1 : 4b 2c 20 53 48 20 36 34  7c
1be9 : 27 45 52 0d 0d 55 4e 50  3e
1bf1 : 41 43 4b 49 4e 47 8e 08  39
1bf9 : 20 2e 2e 20 4d 4f 4d 50  e5
1c01 : 4c 53 20 2e 2e 00 ff ff  a8
```

Listing 1. (Schluß)

Auf der Suche nach der Grafik

Egal, wo Ihre Hires-Grafik liegt, »Hardmaker« findet sie. Multi-Color-Grafiken können auf Wunsch in Graustufen umgerechnet und auf dem MPS 801 und mit den entsprechenden Anpassungen mit 9- oder 24-Nadel-Druckern gedruckt werden.

Der »Hardmaker« erlaubt es, Hires-Grafiken aus fast allen Programmen aufs Papier zu bringen. Dazu stehen umfangreiche Routinen zur Verfügung, die dem Benutzer fast alle Arbeiten abnehmen. Nur das Papier müssen Sie noch von Hand einspannen. Das Programm, das ursprünglich für MPS-Drucker geschrieben war, unterstützt mit zwei Erweiterungen jetzt auch Epson-kompatible 9- oder 24-Nadel-Drucker.

Eingabehinweise

Geben Sie den »Hardmaker« (Listing 1) für den MPS 801/803 bitte mit dem MSE ein, und speichern Sie ihn. Der Programmstart erfolgt durch RUN.

Danach sehen Sie auf dem Bildschirm (meistens) ein wüstes Durcheinander von Punkten. Das ist der Bereich von $2000 bis $4000 als Multicolor-Grafik dargestellt. Das könnte z.B. Teil eines Computerspiels sein, das vorher im Computer war. Wenn Sie nun ein Programm auf Bilder untersuchen wollen, müßten Sie im »Diskworkmodus« dieses Programm laden, die Grafik finden, eventuell speichern und ausdrucken. Dazu stehen Ihnen folgende Funktionen zur Verfügung.

Speicherbereiche

Computergrafiken können nur an bestimmten Stellen im Speicher stehen, um vom VIC ausgelesen werden zu können. Ein solcher Bereich ist der von $2000 bis $3FFF. Diesen Bereich sehen Sie grundsätzlich auf dem Bildschirm; er wird vom Programm als Grafik-RAM benutzt. Wollen Sie den Inhalt eines anderen Bereiches sehen, muß er nach $2000 transportiert werden. Dazu dienen die Tasten <1> bis <6> und <←>:

<1> : $4000 bis $5FFF
<2> : $6000 bis $7FFF
<3> : $8000 bis $9FFF
<4> : $A000 bis $BFFF (RAM unterm Basic)
<5> : $C000 bis $DFFF ($D000 bis $DFFF; RAM unter I/O)
<6> : $E000 bis $FFFF (RAM unterm Kernel)
<←> : $0000 bis $1FFF. Dieser Bereich ist nur der Vollständigkeit halber per Taste erreichbar. Benutzen können Sie ihn nicht, da dort Zeropage, Stack, Video-RAM und der Hardmaker selbst liegen!

Wenn Sie auf eine dieser Tasten ohne <SHIFT>, <CTRL> oder <CBM> drücken, wird der entsprechende Speicherbereich nach $2000 transportiert und ist damit auf dem Bildschirm sichtbar. Drücken Sie jedoch <SHIFT> und eine dieser Tasten, wird der entsprechende Bereich mit dem ab $2000 ODER-verknüpft. So können zwei Bilder zusammengemischt werden. Das Ergebnis liegt wieder ab $2000 im Speicher.

Folgende Kombinationen bewirken also:

<←> bis <6>	≙ jeweiligen Bereich nach $2000 kopieren
<SHIFT>	≙ ODER-Verknüpfen
<CBM>	≙ EX-OR-Verknüpfen
<SHIFT CBM>	≙ UND-Verknüpfen
<CTRL>	≙ Bereich mit dem ab $2000 vertauschen
<CTRL SHIFT> oder <CTRL CBM>	≙ $2000 bis $3FFF in entsprechenden Bereich kopieren

Wenn Sie eine Kombination mit <CTRL> drücken, wird der entsprechende Speicherplatz verändert! Auf diese Weise können Sie z.B. den Inhalt von $2000 bis $3FFF zwischenspeichern, wenn Sie ihn danach weiterbearbeiten wollen (z.B. bei schwierigen Korrekturen).

Bilder »schneiden«

Manchmal kommt es vor, daß nicht der gesamte Inhalt des Bildschirms zu einer Grafik gehört und man den Rest »wegschneiden« möchte. Z.B. wollen Sie am rechten Rand etwas entfernen. Dazu drücken Sie die Taste <R>. Der Rahmen wechselt seine Farbe und an der rechten Seite erscheint eine flackernde Linie, die Sie mit <CURSOR-links/-rechts> hin- und herbewegen können.

Wenn Sie die Linie richtig positioniert haben, drücken Sie auf <SPACE>; der Bereich rechts der Linie wird gelöscht (oder gefüllt, wenn Sie <SHIFT SPACE> drücken). Möchten Sie nichts löschen, dann drücken Sie <Q>, und die flackernde Linie ist verschwunden.

<R> :	rechts	Randfarbe: orange
<L> :	links	Randfarbe: blau
<O> :	oben	Randfarbe: hellrot
<U> :	unten	Randfarbe: grün

Im Schneidemodus stehen folgende Funktionen zur Verfügung:
- Cursor nach oben/unten (nur <O>, <U>)
- Cursor nach oben links/rechts (nur <L>, <R>)
- <SPACE>, <SHIFT SPACE>
- <Q> (wie quit)

Bilder verschieben

Wenn die Grafik nicht genau oben links beginnt, muß sie verschoben werden. Eine Möglichkeit ist das Scrollen:

Mit CBM-Taste + Cursor-Taste wird die Grafik um 1 Byte nach links oder rechts verschoben.

Die andere Möglichkeit sind die Tasten <A> und <E>. Positionieren Sie den Cursor irgendwo mitten auf dem Bildschirm und drücken Sie <A>: Die Grafik wird so verschoben, daß die Cursorposition nun den Anfang der Grafik bildet. Analog funktioniert hier die Taste <E>: Die Cursorposition bildet jetzt das Ende der Grafik. Diese Funktionen ermöglichen ein bequemes Positionieren einer Bitmap, die irgendwo im Speicher liegt.

Farbe

Mit den Funktionstasten kann die Farbe der Grafik geändert werden, wenn die Ausgangsbelegung (die sich für Schwarzweiß-Fernseher übrigens gut eignet) nicht gefällt:

<F1>/<F2>	≙	Farbe 1 Hires/Multi
<F3>/<F4>	≙	Farbe 2 Hires/Multi
<F5>/<F6>	≙	Farbe 3 Multi
<F7>/<F8>	≙	Farbe 4 Multi

Die Funktionstaste blättert die Farben vorwärts, geSHIFTet blättert die Farben zurück. Für Multi und Hires sind getrennte Farb-Speicher vorhanden.

Folgende Tasten bewegen den Cursor:

Cursortasten	gewohnte Cursor-Bewegung
<RETURN>	bewegt ihn in die erste Spalte der nächsten Zeile
<SHIFT CLEAR/HOME>	löscht die Grafik
<CLEAR/HOME>	Cursor links oben
<£>	(Gegenteil von <HOME>: Cursor in letzte Spalte letzte Zeile)
<SPACE>	Cursor rechts + Cursorfeld löschen
<SHIFT SPACE>	Cursor rechts + Cursorfeld füllen mit Farbe 3
<CBM SPACE>	Cursor rechts + Cursorfeld füllen mit Farbe 1
<CTRL SPACE>	Cursor rechts + Cursorfeld füllen mit Farbe 2
<INST/DEL>	genauso wie <SPACE>, bloß mit Cursor nach links (also kein echtes DEL!)

Kurzinfo: Hardmaker

Programmart: Hardcopy-Routine für Hires-Grafiken.
Laden: LOAD "HARDMAKER",8
Start: Nach dem Laden RUN eingeben.
Besonderheiten: Mit dem Zusatzprogramm »MULTIPRINT« wird das Programm an 9-Nadel-Drucker, mit »HARDMAKER 24.OBJ« an 24-Nadel-Drucker angepaßt. Beachten Sie bitte die genauen Installationshinweise am Ende des Artikels.
Programmautor: C. Kurts

Sonderfunktionen

<H>	schaltet Hires-Modus ein; genauso wie Sie die Grafik nun sehen, wird sie von einem Matrixdrucker ausgegeben werden.
<M>	schaltet Multicolor an.
<T>	ist eine sehr praktische Sache, wenn man einen der unter »Hardcopy« erwähnten Matrixdrucker besitzt. Diese geben die Multicolor-Grafiken nämlich so aus, daß die Farben 1 und 2 als charakteristische Linien erscheinen. <T> verwandelt diese Farben nun in Graustufen, die der Drucker ausgeben kann (siehe auch Textkasten). <T> funktioniert nur, wenn Multicolor eingeschaltet ist, und schaltet dann auf Hires um!
<I>	invertiert die Grafik
<S>	spiegelt die Grafik an der Vertikalen, und vertauscht die Farben 1 und 2, die ja ebenfalls gespiegelt werden. Spiegeln an der Horizontalen ist nicht nötig, da man ja die Hardcopy einfach umdrehen kann!
<X>	kehrt ins Basic zurück.

Funktionsweise des T-Befehls beim Hardmaker

Im Multicolor-Modus stellen immer je 2 Bit eines Bytes die Information für einen Bildpunkt zur Verfügung, so daß die Auflösung gegenüber dem Hires-Modus halbiert wird.

Farbige Grafiken in Graustufen umsetzen

Dafür hat man vier Farben statt zwei zur Auswahl, die durch die Bitkombinationen 00, 01, 10 und 11 repräsentiert werden. Ein normaler Matrixdrucker hat hingegen keinen Multicolor-Modus und stellt die Bitmuster genauso dar, wie sie im Grafik-ROM stehen. Das führt bei Flächen, die aus den Farben 1 und 2 bzw. den Bitmustern 01 oder 10 bestehen, zu den bekannten Streifenmustern. Um diese in Graustufen umzuwandeln, geht der Computer folgendermaßen vor: Die vier 2-Bit-Kombinationen eines jeden der 8000 Byte des Grafik-RAMs werden nach den Bitkombinationen 01 und 10 untersucht.

1. Fall: 2-Bit-Kombination = 00 oder 11: nichts wird verändert.

2. Fall: 2-Bit-Kombination = 01: In diesem Fall werden die 2 Bit invertiert, also durch 10 ersetzt, aber nur dann, wenn die 2 Bit Bestandteil des 1., 3., 5. etc. Bytes des Grafik-RAMs sind. Dadurch wird das Bitmuster in jeder zweiten Grafikzeile um 1 Bit nach links versetzt. Wenn die ersten 3 Byte also

```
01 01 00 01                          10 10 00 10
01 01 11 01 lauten, so wird daraus   01 01 11 01
01 01 01 01                          10 10 10 10
```

und der Drucker gibt ein gleichmäßig graues Feld aus.

3. Fall: Bit-Kombination = 10

Hierbei würde aus den ersten 3 Byte

```
10 10 00 10                          01 10 00 10
10 10 11 10 folgendes werden:        10 01 11 01
10 10 10 10                          01 10 01 10
```

Also ein etwas gröberes Graufeld. Das wird dadurch erreicht, daß eine 2-Bit-Kombination invertiert wird, wenn sie entweder im 1., 3., 5., ... Byte die 1. oder 3. 2-Bit-Kombina-

tion oder im 2., 4., 6., ... Byte die 2. oder 4. 2-Bit-Kombination ist.

Diskworkmodus

Wird mit <D> aktiviert. Die Grafik wird dann ausgeblendet und Sie haben den normalen Kernel-Editor vor sich. Folgende Befehle wurden hier implementiert:

$	Directory
@	Kommandokanal des Laufwerks abfragen
@i	initialisieren (analoges gilt für SCRATCH, RENAME, FORMAT etc.)
n	gibt die aktuelle Geräteadresse aus (voreingestellt 8)
n9	schaltet auf Geräteadresse 9 um
?	arbeitet wie der normale Basic-PRINT-Befehl, kann aber hier für Berechnung genutzt werden
Q	schaltet den Grafikmodus wieder an
X	Rückkehr zum Basic
←	Mit ← kann aus dem Hardmaker heraus eine Grafik gespeichert werden; z.B. »←Bild 1 <RETURN>« speichert den Bereich $2000 bis $3FFF (genauer bis $3F40) auf Diskette unter dem Namen »Bild 1« ab. Das erzeugte Programmfile ist 32 Blöcke lang.
£	Um ein Programm nach Grafiken zu durchsuchen, muß man es irgendwie in den Computer bekommen. Die eine Möglichkeit ist, das Programm zu laden, zu starten, mit Reset auszusteigen und dann den Hardmaker nachzuladen. Dies empfiehlt sich immer dann, wenn das Programm die Grafik erst erzeugt. Außerdem liegt diese dann meist an der richtigen Stelle im Speicher, so daß man nicht mehr zu verschieben braucht. Die andere Möglichkeit ist, erst den Hardmaker zu laden und dann mit dem £-Befehl das betreffende Programm (mit £ Name <RETURN>). Dies wird dann in den Bereich ab $2000 geladen, so daß ein Autostart entfällt. Diese Methode kann zu Problemen führen, wenn das betreffende File länger als 178 Blöcke ist, weil dann in die Register der I/O-Bausteine geladen wird. Diese Methode ist aber dann nötig, wenn ein Programm sich nicht durch Reset stoppen läßt.

Hardcopy

Dieser Programmteil ist der wichtigste wie auch der problematischste. Eine Hardcopy-Routine muß an die meisten Drucker speziell angepaßt werden. Die im »Hardmaker« integrierte Hardcopy-Routine spricht Drucker mit sieben Nadeln an, also den MPS 801, den MPS 803, und Epson-Drucker mit entsprechendem Interface. Aktiviert wird die Hardcopy mit der Taste <P>. Daraufhin färbt sich der Rahmen schwarz, und der C 64 beginnt mit der Hardcopy. Danach wird in den Grafikmodus zurückgesprungen.

Warum soll man sich mit einem Epson-Drucker auf die Grafikfähigkeiten eines MPS 801/803 beschränken? Die grafischen Möglichkeiten eines Epson-kompatiblen Druckers erlauben es außerdem, Multicolor-Grafiken durch Zuordnung von Graustufen auszugeben. Dadurch kann eine noch realistischere Wiedergabe in Graustufen (»T-Befehl«) geboten werden.

Für die Ausgabe von Hires-Grafiken wurde bei »Multiprint« darauf Wert gelegt, daß sowohl Bilder im Hires-Modus als auch Grafiken im Multicolor-Modus gedruckt werden können. Um eine möglichst große Flexibilität zu erreichen, wurden Hardcopy-Routinen gewählt, die sowohl Drucker über den seriellen IEC-Bus als auch über eine Centronics-Schnittstelle bedienen können.

An der Bedienung hat sich nichts geändert. Der Start des Druckerprogramms erfolgt durch Drücken der Taste <P>. Dabei erfolgt der Ausdruck im Hires-Modus, wenn die Bildschirm-Darstellung ebenfalls Hires darstellt. Wurde zuvor mit Taste <M> Multicolor angewählt, so wird eine Umsetzung der Bildschirm-Darstellung in Graustufen durchgeführt. Dabei wird der Helligkeitseindruck des momentan angezeigten Bildes zugrundegelegt. Die Parameter für die Hardcopy ermittelt das Programm automatisch. Jeder der 16 möglichen Farben ist über eine Tabelle der Wert 0 (weiß), 1 (hellgrau), 2 (dunkelgrau) und 3 (schwarz) zugeordnet. Aus diesem Grund empfiehlt es sich, das Bild auf einem monochromen Monitor oder einem Schwarzweiß-Fernseher zu betrachten (Farbmonitor: Farbintensität zurückdrehen).

Hinweise zum Eintippen und zur Drucker-Anpassung

Geben Sie das neue Druckprogramm (Listing 2) bitte mit dem MSE ein und speichern Sie es. Laden Sie dann den »Hardmaker«. Danach laden Sie das neue Druckprogramm mit LOAD »MULTIPRINT«,8,1. Jetzt müssen Sie noch die Zeiger für den Programmbeginn und das neue Programm-Ende korrigieren. Dies erreichen Sie mit folgenden POKE-Befehlen im Direktmodus:

```
POKE 43,1:POKE 44,8:POKE 45,30:POKE 46,23
```

In der abgedruckten Form arbeitet das Programm mit Druckern zusammen, die am seriellen Port des Commodore 64 angeschlossen und über Geräteadresse 4, Sekundäradresse 1, im Linearmodus betrieben werden. Die Geräteadresse ist im Programm in der Speicherstelle 5866 ($16EA) abgelegt. Die Sekundäradresse steht in Speicherstelle 5867 ($16EB).

Wollen Sie dagegen einen Drucker mit Parallelschnittstelle direkt am User-Port betreiben, so geben Sie bitte »POKE 5410,0« ein. Dadurch wird der ebenfalls im Programm enthaltene Centronics-Druckertreiber aktiviert.

Wenn Sie noch ein übriges tun wollen, so sollten Sie die Voreinstellungen für die Multicolor-Farben ändern, damit Sie auch auf einem Farbmonitor eine Darstellung in Grauwerten erhalten. Sie müssen dazu lediglich zwei POKE-Befehle geben:

```
POKE 2088,1:POKE 2100,252
```

Jetzt speichern Sie die neue Version mit SAVE "HARDMAKER",8.

Auch für 24-Nadel-Drucker

Listing 3 paßt auf ähnliche Art und Weise den Hardmaker an 24-Nadel-Drucker an. Grafiken kann ein Plotter kaum besser zeichnen. Mit diesen hochwertigen Druckern nutzt man seinen C64 erst richtig aus.

Für die Installation der Hardcopy-Routine gehen Sie folgendermaßen vor:

1. Geben Sie Listing 1 »HARDMAKER 24.OBJ« mit dem MSE ein und speichern das Programm auf Diskette.
2. Laden Sie den Hardmaker wie gewohnt:
   `LOAD "HARDMAKER",8`
3. Laden Sie die neue Hardcopy-Routine HARDMAKER 24.OBJ absolut in den Speicher:
   `LOAD "HARDMAKER 24.OBJ",8,1`
4. Speichern Sie nun den Hardmaker auf Diskette:
   `SAVE"HARDMAKER 24",8`

Nun ist die Installation beendet. Die Bedienung der modifizierten Version ist mit der alten Version völlig identisch. Der Druckvorgang wird nach wie vor mit <P> gestartet. Anschließend färbt sich der Bildschirmrahmen schwarz und das Drucken beginnt...

Ein paar Tricks sollen Ihnen nun die Anpassung an Ihren Drucker, falls notwendig, erleichtern.

Die Druckroutine arbeitet auf allen gängigen 24-Nadel-Druckern, die mit »ESC * 39« die Grafikpunktdichte 180 dpi (Dots per Inch = Punkte pro Zoll) ansteuern. Für ein korrektes Untereinanderliegen der einzelnen Druckzeilen sorgt der »ESC 3 n = 24«-Befehl, der den Zeilenabstand auf 24/180 Zoll festlegt. Weiterhin ist die Hardcopy so programmiert, daß Interface-Fehler, wie sie unter anderem das Merlin-Face C verursacht, ausgeschlossen sind. Das beruht auf der Tatsache, daß die teuflische Fehler verursachenden Bitkombinationen nicht mehr vorkommen. Für einen korrekten Ausdruck muß natürlich die Sekundäradresse auf Linearkanal geschalten werden, damit die Daten unverändert das Interface passieren. Voreingestellt ist die Sekundäradresse 1. Benötigen Sie eine andere Adresse, dann gehen Sie wie folgt vor:

1. Laden Sie den für 24-Nadler angepaßten Hardmaker:
   `LOAD"HARDMAKER 24",8`
2. Geben Sie folgenden Befehl ein:
   `POKE 5029, Sekundäradresse`
3. Für die Einstellung des linken Randes ist
   `POKE 5094, Anzahl der Space am linken Rand`
   zuständig. Speichern Sie nun die an Drucker und Interface angepaßte Hardmaker-Version mit: SAVE"HARDMAKER 24.SA",8.

Mit Hilfe des linken Randes kann man gleichzeitig festlegen, ob ein Linefeed vom Drucker gesendet werden muß. Wird vom Drucker nach »CR« (Carriage Return, CHR$(13)) zusätzlich in »LF« (Linefeed, CHR$(10)) benötigt, genügt es, zum Wert des linken Randes 80 hinzuaddieren.

Funktionsweise: Vor jeder Grafiksequenz wird eine Anzahl von Space zum Drucker gesandt. Übersteigt diese Anzahl Leerzeichen den Wert 80, führt der Drucker einen Wagenrücklauf samt Zeilenvorschub aus. Anschließend druckt er den Rest der Spaces, die nun den linken Rand charakterisieren, und die Grafikzeile. Vorsicht! Der Wert des linken Randes darf nicht Null sein, da der Computer sie als 256 interpretieren würde. Der kleinste Wert für den linken Rand ist 1. (C. Kurts/M. Wilhelm/T. Lipp/ef)

```
Name : hardmaker              0801 1500
----------------------------------------
0801 : 25 08 c0 07 9e 20 32 30  4f
0809 : 38 37 20 20 09 0e 05 57  ad
0811 : 52 49 54 54 45 4e 20 42  73
0819 : 59 20 43 48 52 2e 4b 55  cb
0821 : 52 54 53 00 00 00 a9 0f  37
0829 : 85 fc 8d de 03 a0 00 8c  ba
0831 : dd 03 a9 cb 85 fb a9 20  93
0839 : 8d e0 03 a9 00 8d df 03  1e
0841 : a9 e8 8d 00 03 a9 10 8d  9b
0849 : 01 03 a9 08 8d e2 03 a9  86
0851 : 80 85 9d ad 11 d0 29 bf  6d
0859 : 09 38 8d 11 d0 ad 16 d0  78
0861 : 09 18 8d 16 d0 ad 18 d0  19
0869 : 09 08 8d 18 d0 20 4d 09  32
0871 : a9 13 20 d2 ff a9 0f 8d  ab
0879 : 20 d0 20 14 0b 85 40 8a  7f
0881 : 18 69 20 85 41 a2 02 a0  79
0889 : 07 b1 40 49 ff 91 40 88  41
0891 : 10 f7 ca d0 f2 a5 c6 d0  83
0899 : 09 a6 cb e8 f0 dc a2 0d  a0
08a1 : d0 28 78 20 b4 e5 a2 24  f5
08a9 : dd de 08 d0 14 48 8a 0a  d3
08b1 : a8 b9 04 09 8d c1 08 b9  d3
08b9 : 03 09 8d c0 08 68 4c 1e  ed
08c1 : 0c e0 0f f0 05 ca 10 e0  c8
08c9 : 30 ab a4 cb b9 81 eb cd  64
08d1 : e1 03 d0 04 a9 ff d0 eb  9e
08d9 : 8d e1 03 d0 e6 5f 31 31  c2
08e1 : 32 33 34 35 36 85 86 87  19
08e9 : 88 20 14 4d 48 58 93 13  ec
08f1 : 49 53 41 45 c5 5c 0d 11  72
08f9 : 1d 91 9d 4c 52 4f 55 44  4d
0901 : 50 54 82 09 82 09 82 09  ca
0909 : 82 09 82 09 82 09 82 09  5e
0911 : 82 09 6c 0b a3 0b cb 0b  6c
0919 : e3 0b fb 0b fb 0b 1f 0d  91
0921 : 36 0d 1e 0c 5d 0c 4d 0b  68
0929 : 6c 0c 89 0c c1 0a aa 0a  aa
0931 : aa 0a 62 0b 53 0b 47 0b  9b
0939 : 41 0b 2d 0b 37 0b 51 0d  d8
0941 : 49 0d ea 0e f2 0e 8e 10  87
0949 : 74 13 0a 13 a2 00 ad de  ca
0951 : 03 8d 21 d0 ad 16 d0 29  9e
0959 : 10 f0 03 a5 fb 2c a5 fc  08
0961 : 9d 00 04 9d 00 05 9d 00  52
0969 : 06 9d 00 07 48 ad dd 03  8e
0971 : 9d 00 d8 9d 00 d9 9d 00  3d
0979 : da 9d 00 db 68 e8 d0 e0  70
0981 : 60 ac 8d 02 a9 3f 85 3c  fe
0989 : bd ea 09 85 8c 48 38 e9  6e
0991 : 1f 8d e0 03 a9 00 8d df  a0
0999 : 03 68 c0 04 f0 2f 90 0e  68
09a1 : ca 30 1e 85 3c a9 01 8d  ec
09a9 : 20 d0 a9 3f 85 8c b9 f2  0d
09b1 : 09 8d c7 09 a9 c5 8d 0d  ad
09b9 : 0a a9 09 8d 0e 0a 20 fa  34
09c1 : 09 4c 76 08 b1 8b 64 3b  0e
09c9 : 91 3b 4c 0f 0a ca 30 f1  88
09d1 : a9 dd 8d 0d 0a a9 09 8d  9b
09d9 : 0e 0a d0 e2 b1 8b 48 b1  79
09e1 : 3b 91 8b 68 91 3b 4c 0f  17
09e9 : 0a 1f 3f 5f 7f 9f bf df  f2
09f1 : ff 64 11 51 31 65 44 04  e8
09f9 : d4 78 a9 34 85 01 38 a5  87
0a01 : 3c e9 20 85 40 a0 00 84  fd
0a09 : 8b 84 3b 4c c5 09 88 d0  97
0a11 : fa c6 8c a5 40 c7 3c d0  1b
0a19 : f2 a9 37 85 01 58 60 a9  06
0a21 : ff cf df 03 d0 03 ce e0  82
0a29 : 03 a9 1f 85 60 a2 f8 86  85
0a31 : 5f a2 3f 86 5b 86 59 a2  17
0a39 : 47 86 5a e8 86 58 20 bf  a2
0a41 : a3 a9 00 8d f8 1f 4c 76  11
0a49 : 08 ee df 03 d0 03 ee e0  c3
0a51 : 03 a9 1f 85 60 85 59 a2  7e
0a59 : f8 86 58 e8 86 5f a2 3f  34
0a61 : 86 5b a2 48 86 5a 20 71  e5
0a69 : 0a a9 00 8d 47 3f f0 d6  d9
0a71 : 38 a5 5a e5 5f 85 fd 49  7c
0a79 : ff a8 a5 5b e5 60 aa e8  7f
0a81 : 18 a5 5f 65 fd 85 5f a5  c5
0a89 : 60 e9 00 85 60 18 a5 58  9d
0a91 : 65 fd 85 58 a5 59 e9 00  2e
0a99 : 85 59 b1 5f 91 58 c8 d0  c4
0aa1 : f9 e6 60 e6 59 ca d0 f2  17
0aa9 : 60 20 14 0b a0 27 84 8b  ec
0ab1 : a0 18 84 8c 38 e9 38 48  54
0ab9 : 8a e9 1f aa 68 4c ca 0a  7d
0ac1 : 20 14 0b a0 00 84 8b 84  1e
0ac9 : 8c 18 6d df 03 8d df 03  db
0ad1 : 85 5a 85 5f 8a 6d e0 03  6e
0ad9 : 8d e0 03 85 60 18 69 20  f5
0ae1 : 85 5b a9 00 85 58 a9 20  80
0ae9 : 85 59 ad e0 03 c9 20 90  c2
0af1 : 0b 78 a9 34 85 01 20 71  ed
0af9 : 0a 4c 04 0b a9 40 85 59  f1
0b01 : 20 bf a3 a9 37 85 01 58  73
0b09 : a6 8c a4 8b 18 20 f0 ff  d6
0b11 : 4c 50 0b a5 d1 18 65 d3  18
0b19 : aa a5 d2 e9 03 85 fe 8a  f5
0b21 : 0a 26 fe 0a 26 fe 0a 26  0e
0b29 : fe a6 fe 60 ae 8d 02 e0  67
0b31 : 02 90 19 4c 4a 0a ae 8d  16
0b39 : 02 e0 02 90 0f 4c 20 0a  26
0b41 : a6 d3 e0 27 d0 06 a6 d6  73
0b49 : e0 18 f0 03 20 d2 ff 4c  03
0b51 : 76 08 a6 d6 e0 18 d0 f4  4c
0b59 : a9 91 20 d2 ff a9 0d d0  50
0b61 : eb a0 27 a2 18 18 20 f0  5f
0b69 : ff d0 e4 a9 e8 ac 8d 02  6d
0b71 : f0 02 a9 ca 8d 92 0b ad  1b
0b79 : 16 d0 29 10 f0 03 a9 fb  09
0b81 : 2c a9 fc 8d 8e 0b 8d 98  1b
0b89 : 0b 8d 9c 0b a5 01 29 0f  08
0b91 : aa 9a 8a 09 f0 aa a5 01  49
0b99 : 09 0f 87 00 20 4d 09 4c  35
0ba1 : 76 08 a9 10 ac 8d 02 f0  a9
0ba9 : 02 a9 f0 8d c5 0b ad 16  05
0bb1 : d0 29 10 f0 03 a9 fb 2c  fe
0bb9 : a9 fc 8d c2 0b 8d c7 0b  ee
0bc1 : a5 fb 18 69 10 85 fb 4c  4d
0bc9 : 9d 0b ad 16 d0 29 10 f0  92
0bd1 : cb ac 8d 02 d0 06 ee dd  4b
0bd9 : 03 4c 9d 0b ce dd 03 4c  4c
0be1 : 9d 0b ad 16 d0 29 10 f0  aa
0be9 : b3 ac 8d 02 d0 06 ee de  4d
0bf1 : 03 4c 9d 0b ce de 03 4c  6c
0bf9 : 9d 0b ac 8d 02 b9 16 0c  57
0c01 : a0 07 91 40 88 10 fb e0  4c
0c09 : 0d f0 05 a9 1d 4c 41 0b  54
0c11 : a9 9d 4c 4d 0b 00 ff 55  a1
0c19 : aa 3c cc 33 c3 20 30 0c  91
0c21 : a9 8b 8d 00 03 a9 e3 8d  1b
0c29 : 01 03 68 68 4c 7b e3 a9  56
0c31 : 00 85 c6 20 44 e5 ad 11  f6
0c39 : d0 29 df 8d 11 d0 a9 06  92
0c41 : 8d 21 d0 ad 16 d0 29 ef  b5
0c49 : 8d 16 d0 a9 0e 8d 20 d0  ba
0c51 : 8d 86 02 ad 18 d0 29 f7  f4
0c59 : 8d 18 d0 60 ac 8d 02 f0  54
0c61 : 07 a0 02 a2 01 4c 85 09  28
0c69 : 4c 76 08 a9 3f 85 3c a9  8c
0c71 : 80 8d 0d 0a a9 0c 8d 0e  8a
0c79 : 0a 20 fa 09 4c 76 08 b1  6f
0c81 : 3b 49 ff 91 3b 4c 0f 0a  f9
0c89 : a9 a0 85 8b a9 98 85 3b  41
0c91 : a9 3e 85 3c 85 8c a9 14  ce
0c99 : 48 a0 07 b1 8b 85 fe b1  6e
0ca1 : 3b aa 0a 8a a2 08 66 fe  07
0ca9 : 2a ca d0 fa 91 3b a5 fe  53
0cb1 : 6a 91 8b 88 10 e5 68 aa  ff
0cb9 : ca f0 1e 8a 48 a5 8b 18  e4
0cc1 : 69 08 85 8b a5 8c 69 00  65
0cc9 : 85 8c a5 3b 38 e9 08 85  63
```

```
0cd1 : 3b a5 3c e9 00 85 3c d0  ea
0cd9 : c0 a5 3b 38 e9 a8 85 3b  b2
0ce1 : 85 8b aa a5 3c e9 00 85  a9
0ce9 : 3c 85 8c a8 8a 18 69 08  3f
0cf1 : 85 8b 98 69 00 85 8c c9  81
0cf9 : 1f d0 9b ad 16 d0 29 10  ca
0d01 : f0 23 a5 fb 48 29 f0 4a  92
0d09 : 4a 4a 4a 85 fe 68 29 0f  b2
0d11 : 0a 0a 0a 0a 05 fe 85 fb  3b
0d19 : 20 4d 09 4c 76 08 ad 16  36
0d21 : d0 29 10 f0 03 4c 76 08  24
0d29 : ad 16 d0 09 10 8d 16 d0  9e
0d31 : 20 4d 09 f0 f0 ad 16 d0  cf
0d39 : 29 10 d0 02 f0 e7 ad 16  10
0d41 : d0 29 ef 8d 16 d0 d0 e8  50
0d49 : a9 27 a2 01 a0 08 d0 0f  fa
0d51 : a2 80 ad 16 d0 29 10 f0  da
0d59 : 02 a2 40 a9 00 a0 06 85  1a
0d61 : 8b 86 8c 8c 20 d0 20 bf  6c
0d69 : 0e 20 bf 0e a5 c6 d0 05  17
0d71 : e6 c6 20 b3 ee 78 20 b4  d5
0d79 : e5 c9 9d f0 13 c9 1d f0  9e
0d81 : 3a c9 20 f0 5a c9 a0 f0  1e
0d89 : 59 c9 51 d0 d9 4c 76 08  1f
0d91 : ad 16 d0 29 10 d0 12 a5  be
0d99 : 8b d0 04 a6 8c 30 c7 06  d8
0da1 : 8c 90 c3 26 8c c6 8b 10  78
0da9 : bd a5 8b d0 06 a5 8c c9  89
0db1 : 40 b0 b3 06 8c 06 8c 90  43
0db9 : ad b0 e8 a5 8b c9 27 90  72
0dc1 : 05 a6 8c ca f0 a0 ad 16  8d
0dc9 : d0 29 10 d0 0a 46 8c 90  72
0dd1 : 95 66 8c e6 8b 10 8f 46  9d
0dd9 : 8c 66 8c 10 89 30 f2 a9  f7
0de1 : 00 2c a9 ff 85 fd ad 20  a1
0de9 : d0 29 0f c9 08 f0 5e a9  20
0df1 : 20 85 41 a9 00 85 40 a5  d2
0df9 : 41 85 3c a5 40 85 3b a6  2b
0e01 : 8b f0 17 a0 07 a5 fd 91  97
0e09 : 3b 88 10 fb 18 a9 08 65  c6
0e11 : 3b 85 3b 90 02 e6 3c ca  ce
0e19 : d0 e9 a6 8c ca 8a a2 29  f7
0e21 : a4 fd f0 04 49 ff a2 09  b2
0e29 : a0 07 8d 34 0e 8e 33 0e  75
0e31 : b1 3b 49 00 91 3b 88 10  07
0e39 : f7 18 a5 40 69 40 85 40  dd
0e41 : a5 41 69 01 85 41 c9 3f  09
0e49 : 90 ad 4c 76 08 a9 21 85  ef
0e51 : 41 a9 38 85 40 a5 40 85  63
0e59 : 3b a5 41 85 3c 38 a9 27  e2
0e61 : e5 8b aa f0 19 a0 07 a5  d2
0e69 : fd 91 3b 88 10 fb 38 a5  1c
0e71 : 3b e9 08 85 3b a5 3c e9  f9
0e79 : 00 85 3c ca d0 e7 a5 8c  a0
0e81 : ae 16 d0 e0 10 90 01 0a  28
0e89 : 85 fe 38 e9 01 05 fe a2  52
0e91 : 09 a4 fd d0 04 49 ff a2  56
0e99 : 29 a0 07 8d a5 0e 8e a4  d4
0ea1 : 0e b1 3b 49 00 91 3b 88  0a
0ea9 : 10 f7 18 a5 40 69 40 85  cb
0eb1 : 40 a5 41 69 01 85 41 c9  16
0eb9 : 40 90 9a 4c 76 08 a5 8b  c7
0ec1 : 0a 0a 0a 85 40 a9 00 69  28
0ec9 : 20 85 41 a2 19 a0 07 b1  67
0ed1 : 40 45 8c 91 40 88 10 f7  82
0ed9 : 18 a5 40 69 40 85 40 a5  7e
0ee1 : 41 69 01 85 41 ca 10 e5  3e
0ee9 : 60 a2 20 a0 0a a9 00 f0  86
0ef1 : 06 a2 3e a0 05 a9 07 85  b1
0ef9 : 8b 86 8c 8c 20 d0 20 27  d3
0f01 : 10 20 27 10 a5 c6 d0 05  cb
0f09 : e6 c6 8d 77 02 78 20 b4  72
0f11 : e5 c9 91 f0 13 c9 11 f0  03
0f19 : 3a c9 20 f0 61 c9 a0 f0  27
0f21 : 60 c9 51 d0 d9 4c 76 08  be
0f29 : a5 8c c9 20 d0 04 a5 8b  66
0f31 : f0 cc a5 8b 29 07 f0 0c  09
0f39 : c6 8b a5 8b c9 ff d0 be  fd
0f41 : c6 8c d0 ba a5 8b 38 e9  44
0f49 : 39 85 8b a5 8c e9 01 85  03
0f51 : 8c d0 ab a5 8c c9 3e 90  16
0f59 : 06 a5 8b c9 07 f0 9f a5  10
0f61 : 8b 29 07 49 07 f0 08 e6  52
0f69 : 8b d0 93 e6 8c d0 8f a5  f7
0f71 : 8b 18 69 39 85 8b a5 8c  ee
0f79 : 69 01 85 8c d0 80 a9 00  0d
0f81 : 2c a9 ff 85 3c ad 20 d0  86
0f89 : 29 0f c9 0a f0 72 a5 8c  40
0f91 : c9 3e b0 24 a5 8b 29 f8  77
0f99 : 18 69 41 85 5f a5 8c 69  8f
0fa1 : 01 85 60 a2 40 86 5a ca  08
0fa9 : 86 58 a9 3f 85 5b 85 59  aa
0fb1 : a5 3c 8d 3f 3f 20 bf a3  fb
0fb9 : a9 28 85 40 a5 8b a2 07  2f
0fc1 : 87 3b 29 f8 85 8b ad 20  fb
0fc9 : d0 29 0f aa a5 3c a0 00  06
0fd1 : c4 3b f0 09 e0 05 f0 02  8e
0fd9 : 91 8b c8 d0 f3 91 8b c8  07
0fe1 : c0 08 b0 09 e0 0a f0 02  19
0fe9 : 91 8b c8 d0 f3 a5 8b 69  f9
0ff1 : 07 85 8b a5 8c 69 00 85  71
0ff9 : 8c c6 40 d0 cf 4c 76 08  5c
1001 : a5 8c c9 21 90 b2 85 5b  ee
1009 : a5 8b 29 f8 85 5a a9 20  ef
1011 : 85 60 85 59 a2 01 86 58  50
1019 : ca 86 5f a5 3c 8d ff 1f  21
1021 : 20 71 0a 4c b9 0f a5 8b  c8
1029 : 85 40 a5 8c 85 41 a0 28  fe
1031 : a2 00 a1 40 49 ff 81 40  5f
1039 : 18 a5 40 69 08 85 40 90  30
1041 : 02 e6 41 88 d0 ec 60 09  20
1049 : 0e 20 c8 41 52 44 43 4f  b5
1051 : 50 59 2d cd 41 4b 45 52  7b
1059 : 20 20 20 57 52 49 54 54  e6
1061 : 45 4e 20 42 59 20 c3 48  54
1069 : 52 2e 20 cb 55 52 54 53  34
1071 : 20 20 20 20 20 20 20 20  71
1079 : 20 20 2d 20 c4 49 53 4b  73
1081 : 57 4f 52 4b 2d cd 4f 44  85
1089 : 55 53 0d 0d 00 20 30 0c  47
1091 : a0 00 b9 48 10 f0 06 20  8a
1099 : d2 ff c8 d0 f5 20 cc ff  4a
10a1 : 20 60 a5 86 7a 84 7b a2  2a
10a9 : 07 20 73 00 c9 00 f0 f0  df
10b1 : dd d0 10 f0 08 ca 10 f8  22
10b9 : a2 11 4c e8 10 8a 0a aa  e7
10c1 : bd d8 10 8d ce 10 bd d9  b8
10c9 : 10 8d cf 10 4c 00 00 4e  f7
10d1 : 24 40 51 58 5f 5c 3f 4c  e3
10d9 : 12 2a 11 c2 11 54 08 21  b3
10e1 : 0c b5 12 70 12 55 12 8a  84
10e9 : 0a aa 20 f4 10 20 7a a6  28
10f1 : 4c a1 10 bd 26 a3 85 22  a3
10f9 : bd 27 a3 85 23 a9 00 85  6e
1101 : 13 20 cc ff 20 3f ab a0  43
1109 : 00 b1 22 48 29 7f 20 d2  28
1111 : ff c8 68 10 f4 a0 00 b9  58
1119 : 6a a3 20 d2 ff c8 c0 06  0c
1121 : 90 f5 20 44 12 20 44 12  94
1129 : 60 20 2a 12 90 06 20 44  a8
1131 : 12 4c a1 10 20 44 12 a5  8b
1139 : 7b 85 bc a5 7a 85 bb 20  5e
1141 : 64 12 84 b7 ad e2 03 85  cf
1149 : ba a9 60 85 b9 20 d5 f3  7c
1151 : a5 ba 20 b4 ff a5 b9 20  46
1159 : 96 ff a9 00 85 90 a0 03  bf
1161 : 84 40 20 a5 ff 85 41 a4  3d
1169 : 90 d0 41 20 a5 ff a4 90  c4
1171 : d0 3a a4 40 88 d0 e9 48  d7
1179 : a0 06 20 3f ab 88 d0 fa  45
1181 : 68 a6 41 20 cd bd 20 3f  5a
1189 : ab 20 3f ab 20 a5 ff a6  06
1191 : 90 d0 19 aa f0 06 20 d2  8a
1199 : ff 4c 8d 11 a5 91 10 0c  83
11a1 : ad 8d 02 d0 fb 20 44 12  a5
11a9 : a0 02 d0 b4 20 44 12 a5  cd
11b1 : 90 29 83 d0 06 20 42 f6  29
11b9 : 4c a1 10 20 e0 11 4c a1  e9
11c1 : 10 20 2a 12 b0 0a 20 73  71
11c9 : 00 c9 00 d0 06 20 e0 11  cf
11d1 : 4c a1 10 c9 24 d0 03 4c  99
11d9 : 2a 11 20 09 12 90 f1 a0  64
11e1 : 00 84 90 ad e2 03 85 ba  cf
11e9 : 20 b4 ff a9 6f 85 b9 20  e3
11f1 : 96 ff 20 a5 ff 24 90 70  88
11f9 : 05 20 d2 ff d0 f4 20 ab  4f
1201 : ff 20 44 12 20 44 12 60  91
1209 : ad e2 03 85 ba 20 b1 ff  0c
1211 : a9 6f 85 b9 20 93 ff 20  e9
1219 : 79 00 20 a8 ff 20 73 00  7e
1221 : c9 00 d0 f6 20 ae ff 18  a5
1229 : 60 ad e2 03 85 ba a2 00  32
1231 : 86 90 20 b4 ff 20 ab ff  4e
1239 : a5 90 f0 eb a2 0a 20 f4  c5
1241 : 10 38 60 48 a9 0d 20 d2  b8
1249 : ff 68 60 20 9b b7 8e e2  10
1251 : 03 4c a1 10 20 79 a5 20  89
1259 : 73 00 20 73 00 20 a0 aa  1c
1261 : 4c a1 10 a0 00 b1 7a f0  ef
1269 : 05 c8 c0 28 90 f7 60 20  92
1271 : 2a 12 b0 34 a9 20 8d e0  eb
1279 : 03 a9 00 8d df 03 20 73  80
1281 : 00 a5 7b 85 bc a5 7a 85  d1
1289 : bb 20 64 12 84 b7 ad e2  32
1291 : 03 85 ba a9 00 85 b9 a2  93
1299 : 00 86 0a a0 20 20 d5 ff  cd
12a1 : b0 09 20 44 12 20 e0 11  2e
12a9 : 4c a1 10 20 e8 12 20 e0  2f
12b1 : 11 4c a1 10 20 2a 12 b0  50
12b9 : ef 20 73 00 a5 7b 85 bc  5b
12c1 : a5 7a 85 bb 20 64 12 84  f3
12c9 : b7 a9 00 ae e2 03 a8 20  54
12d1 : ba ff a9 00 85 69 a9 20  80
12d9 : 85 6a a2 40 a0 3f a9 69  c1
12e1 : 20 d8 ff 90 bd b0 c4 a2  39
12e9 : 00 bd f6 12 f0 06 20 d2  2d
12f1 : ff e8 d0 f5 60 0d 4f 50  a3
12f9 : 45 52 41 54 49 4f 4e 20  cb
1301 : 53 54 4f 50 50 45 44 0d  b7
1309 : 00 ad 16 d0 29 10 f0 11  78
1311 : a9 3f 85 3c a9 25 8d 0d  57
1319 : 0a a9 13 8d 0e 0a 20 fa  16
1321 : 09 4c 3f 0d a2 03 b1 3b  41
1329 : 3d 68 13 f0 34 48 3d 6c  d1
1331 : 13 f0 13 68 3d 70 13 d0  d3
1339 : 28 98 4a b0 24 b1 3b 5d  cd
1341 : 68 13 91 3b d0 1b 68 84  8f
1349 : fd 8a 45 fd 4a b0 09 bd  66
1351 : 68 13 11 3b 91 3b d0 09  37
1359 : bd 68 13 49 ff 31 3b 91  d2
1361 : 3b ca 10 c2 4c 0f 0a c0  45
1369 : 30 0c 03 40 10 04 01 80  8e
1371 : 20 08 02 a9 00 8d 20 d0  da
1379 : 20 a0 13 4c 76 08 ea ea  61
1381 : ea ea ea ea ea ea ea ea  80
1389 : ea ea ea ea ea ea ea ea  88
```

Listing 1. »Hardmaker« erzeugt Hardcopys mit MPS 801/803. Bitte mit dem MSE (Seite 159) eingeben.

```
1391 : ea ea ea ea ea ea 56 f2  4d
1399 : df 2d 00 52 d3 eb e7 a9  e9
13a1 : 20 8d f5 14 a9 04 85 ba  ce
13a9 : a2 00 86 90 86 fe 20 b1  43
13b1 : ff 20 ae ff a6 90 f0 01  21
13b9 : 60 86 b9 86 b7 e8 86 b8  ea
13c1 : 20 c0 ff a6 b8 20 c9 ff  ca
13c9 : a9 ff 85 61 a9 07 8d f6  f6
13d1 : 14 a9 1c 85 97 a9 00 8d  53
13d9 : f1 14 a9 28 8d f3 14 a2  52
13e1 : 04 bd ce 14 20 d2 ff ca  28
13e9 : 10 f7 a9 00 85 63 85 64  b2
13f1 : ad f1 14 85 65 a9 00 8d  0b
13f9 : f7 14 a5 63 a6 64 a4 65  bb
1401 : 20 93 14 ae f7 14 a5 ad  d8
1409 : a0 00 b1 ac ae f7 14 9d  e1
1411 : f8 14 e6 65 e8 8e f7 14  85
1419 : ec f6 14 d0 dd a9 00 a0  0c
1421 : 07 d0 02 d0 b5 ae f6 14  00
1429 : 1e f8 14 2a ca 10 f9 25  6d
1431 : 61 09 80 20 d2 ff ad 8d  3a
1439 : 02 29 01 d0 f9 a5 91 10  5d
1441 : 3d 88 10 e1 a5 63 18 69  ab
1449 : 08 85 63 90 02 e6 64 ce  85
1451 : f3 14 d0 9c a9 0d 20 d2  3f
1459 : ff ad f1 14 18 69 07 8d  32
1461 : f1 14 c6 97 f0 02 d0 bb  db
1469 : a9 04 cd f6 14 f0 0f 8d  87
1471 : f6 14 a9 01 85 97 a9 0f  d6
1479 : 85 61 d0 ea a9 01 85 fe  f7
1481 : a9 0f 20 d2 ff a9 0d 20  d6
1489 : d2 ff 20 cc ff a9 01 4c  e6
1491 : c3 ff 85 14 86 15 98 4a  40
1499 : 4a 4a aa bd d3 14 85 ad  ba
14a1 : 8a 29 03 aa bd ed 14 85  7d
14a9 : ac 98 29 07 18 65 ac 85  37
14b1 : ac a5 14 29 f8 85 63 ad  ff
14b9 : f5 14 18 65 ad 85 ad a5  74
14c1 : ac 18 65 63 85 ac a5 ad  ef
14c9 : 65 15 85 ad 60 50 00 10  79
14d1 : 1b 08 00 01 02 03 05 06  69
14d9 : 07 08 0a 0b 0c 0d 0f 10  4e
14e1 : 11 12 14 15 16 17 19 1a  56
14e9 : 1b 1c 1e 1f 00 40 80 c0  03
14f1 : 00 00 00 00 00 00 00 00  f2
14f9 : 00 00 00 00 00 00 00 ff  f9
```

Listing 1. (Schluß)

```
Name : multiprint           13a0 171e
-------------------------------------
13a0 : a9 00 85 b7 ad ea 16 85  37
13a8 : ba 85 b8 ad eb 16 85 b9  02
13b0 : ad 16 d0 29 10 d0 06 a9  b5
13b8 : 81 4c 80 15 ea a6 b8 20  29
13c0 : c0 ff a6 b8 20 c9 ff a9  e4
13c8 : 20 85 15 a0 00 84 14 ae  d6
13d0 : de 03 20 05 14 98 95 57  e4
13d8 : c8 a5 fb 4a 4a 4a 4a aa  30
13e0 : 20 05 14 98 95 57 c8 a5  1d
13e8 : fb 29 0f aa 20 05 14 98  3d
13f0 : 95 57 c8 ae dd 03 20 05  b9
13f8 : 14 98 95 57 20 1a 14 ea  a2
1400 : a9 01 4c c3 ff bd 0a 14  f4
1408 : aa 60 03 00 03 01 02 02  e8
1410 : 03 01 02 03 02 03 02 01  b7
1418 : 02 01 a0 01 b9 44 15 20  35
1420 : 21 15 88 10 f7 a9 00 85  c8
1428 : 62 a9 00 85 63 20 e4 ff  da
1430 : f0 03 4c 0d 15 a0 03 b9  2c
1438 : 40 15 20 21 15 88 10 f7  f5
1440 : a9 00 85 61 a5 14 a4 15  2f
1448 : 85 22 84 23 a4 63 a2 00  54
1450 : 78 a9 35 85 01 b1 22 95  ec
1458 : 64 c8 e8 e0 04 d0 f6 a9  6c
1460 : 37 85 01 58 a9 03 85 60  2f
1468 : a2 03 b5 64 a4 60 f0 06  a3
1470 : 4a 4a 88 4c 6e 14 18 29  c5
1478 : 03 95 5b ca 10 ec a0 00  61
1480 : a2 03 a9 00 85 5f a9 03  0e
1488 : 85 68 b5 5b c5 57 d0 06  80
1490 : b9 48 15 18 90 17 c5 58  3f
1498 : d0 06 b9 54 15 18 90 0d  d3
14a0 : c5 59 d0 06 b9 60 15 18  2a
14a8 : 90 03 b9 6c 15 25 68 05  dc
14b0 : 5f 85 5f 18 26 68 18 26  ff
14b8 : 68 18 ca 10 cd 20 21 15  6d
14c0 : c8 c0 08 d0 bb a6 60 ca  0c
14c8 : 86 60 10 9c 18 a5 22 69  20
14d0 : 08 85 22 a5 23 69 00 85  61
14d8 : 23 a6 61 e8 86 61 e0 28  0b
14e0 : f0 03 4c 4c 14 a6 63 e8  c4
14e8 : e8 e8 e8 86 63 e0 08 f0  8e
14f0 : 03 4c 2d 14 18 a5 14 69  b9
14f8 : 40 85 14 a5 15 69 01 85  60
1500 : 15 a6 62 e8 86 62 e0 19  4f
1508 : f0 03 4c 29 14 a0 01 b9  70
1510 : 46 15 20 21 15 88 10 f7  d3
1518 : 20 cc ff a9 01 20 c3 ff  f3
1520 : 60 a2 01 f0 03 4c d2 ff  0e
1528 : 8d 01 dd a9 10 2c 0d dd  35
1530 : f0 fb ad 00 dd 09 04 8d  db
1538 : 00 dd 29 fb 8d 00 dd 60  02
1540 : 05 00 4c 1b 31 1b 40 1b  df
1548 : 00 00 00 00 00 00 00 00  49
1550 : 00 00 00 00 aa 00 55 00  50
1558 : aa 00 55 00 aa 00 55 00  58
1560 : aa 00 00 00 55 00 00 00  60
1568 : aa 00 00 00 aa 55 aa 55  bd
1570 : aa 55 aa 55 aa 55 aa 55  c5
1578 : 00 00 00 00 00 00 00 00  79
1580 : 85 5b ad 22 15 d0 22 78  b4
1588 : a9 ff 8d 03 dd ad 02 dd  04
1590 : 09 04 8d 02 dd ad 00 dd  46
1598 : 09 04 8d 00 dd a9 10 8d  8d
15a0 : 0d dd ad 0d dd 58 4c c4  04
15a8 : 15 20 cc ff a9 04 ae ea  4c
15b0 : 16 ac eb 16 20 ba ff a9  05
15b8 : 00 20 bd ff 20 c0 ff a2  85
15c0 : 04 20 c9 ff a9 04 85 5e  d4
15c8 : a0 03 20 a2 16 a9 80 85  02
15d0 : 5f a9 02 85 60 a5 5b 29  28
15d8 : c0 d0 04 46 5f 46 60 a5  bf
15e0 : 5b 29 07 a2 00 20 ba 16  fe
15e8 : a5 5b 30 03 4a 4a 4a 29  1a
15f0 : 07 a2 02 20 ba 16 a9 19  02
15f8 : 85 5c a5 5f 30 0a a0 14  ff
1600 : a9 20 20 21 15 88 d0 fa  b4
1608 : a0 08 20 a2 16 a5 5f 20  55
1610 : 21 15 a5 60 20 21 15 a9  e4
1618 : 28 85 5d 78 a9 34 85 01  be
1620 : a0 07 b1 57 99 1e 17 88  93
1628 : 10 f8 a9 37 85 01 58 a5  13
1630 : 57 18 69 08 85 57 90 02  48
1638 : e6 58 a0 08 a2 00 3e 1e  d3
1640 : 17 08 2a e8 28 24 5b 30  74
1648 : 06 e0 08 d0 f1 f0 08 2a  f6
1650 : e4 5e d0 ea 20 21 15 20  94
1658 : 21 15 88 d0 df c6 5d d0  8b
1660 : ba a5 5f 10 26 a2 01 b4  ac
1668 : 57 b5 59 95 57 94 59 ca  b8
1670 : 10 f5 a5 5e 49 0c 85 5e  78
1678 : c9 04 f0 0f a5 5b 30 03  5d
1680 : 4c 17 16 a0 00 20 a2 16  a9
1688 : 4c fa 15 c6 5c d0 f4 a0  d1
1690 : 00 20 a2 16 a0 0d 20 a2  44
1698 : 16 20 cc ff a9 04 20 c3  b4
16a0 : ff 60 b9 ec 16 c9 ff f0  6d
16a8 : 06 20 21 15 c8 d0 f3 60  4d
16b0 : a0 06 d9 96 1f f0 03 88  33
16b8 : d0 f8 a9 00 95 57 a9 20  6a
16c0 : 95 58 60 fb 95 58 60 ae  14
16c8 : e9 16 d0 03 4c d2 ff 48  3d
16d0 : 20 fd 16 ad 00 dd 29 fb  b6
16d8 : 8d 00 dd 09 04 8d 00 dd  66
16e0 : ad 0d dd 29 10 f0 f9 68  f2
16e8 : 60 01 04 01 0d ff ff 1b  f1
16f0 : 31 ff ff ff 1b 4b ff ff  2d
16f8 : ff 1b 32 ff ff 48 ad 0d  24
1700 : dd a9 08 8d 04 dd a9 00  3b
1708 : 8d 05 dd a9 41 8d 0e dd  39
1710 : 68 8d 0c dd 48 ad 0d dd  df
1718 : 29 08 f0 f9 68 60 ff ff  4a
```

Listing 2. »Multiprint«, eine Erweiterung für Epson-kompatible Drucker

```
Name : hardmaker 24.obj  13a0 14ef
-------------------------------------
13a0 : a9 04 a2 04 a0 01 20 ba  7c
13a8 : ff 20 c0 ff a2 04 20 c9  46
13b0 : ff a9 1b 20 d2 ff a9 40  a3
13b8 : 20 d2 ff a9 0d 20 d2 ff  93
13c0 : a9 1b 20 d2 ff a9 33 20  b4
13c8 : d2 ff a9 18 20 d2 ff a9  f3
13d0 : 0d 20 d2 ff a9 00 8d 10  93
13d8 : 14 a9 20 8d 11 14 a9 19  05
13e0 : 8d ed 14 a9 20 a2 0c 20  26
13e8 : d2 ff ca d0 fa a9 1b 20  30
13f0 : d2 ff a9 2a 20 d2 ff a9  5d
13f8 : 27 20 d2 ff a9 c0 20 d2  ab
1400 : ff a9 03 20 d2 ff a9 28  bd
1408 : 8d ec 14 a0 00 a2 00 b9  ad
1410 : ff ff 0a 3e cc 14 e8 e0  2c
1418 : 08 d0 f7 c8 c0 08 d0 ed  0b
1420 : a2 07 a0 07 bd cc 14 0a  f5
1428 : 3e d4 14 3e dc 14 3e e4  ce
1430 : 14 bd cc 14 0a 3e d4 14  e7
1438 : 3e dc 14 3e e4 14 bd cc  31
1440 : 14 0a 3e d4 14 3e dc 14  52
1448 : 3e e4 14 9d cc 14 88 10  61
1450 : d3 ca 10 ce a2 00 bd e4  51
1458 : 14 20 d2 ff bd dc 14 20  84
1460 : d2 ff bd d4 14 20 d2 ff  c9
1468 : bd e4 14 20 d2 ff bd dc  7e
1470 : 14 20 d2 ff bd d4 14 20  5c
1478 : d2 ff bd e4 14 20 d2 ff  e3
1480 : bd dc 14 20 d2 ff bd d4  82
1488 : 14 20 d2 ff e8 e0 08 d0  b8
1490 : c5 18 ad 10 14 69 08 8d  97
1498 : 10 14 90 03 ee 11 14 ce  9c
14a0 : ec 14 f0 03 4c 0b 14 a9  f4
14a8 : 0d 20 d2 ff ce ed 14 f0  08
14b0 : 03 4c e3 13 a9 1b 20 d2  ce
14b8 : ff a9 40 20 d2 ff a9 0d  8e
14c0 : 20 d2 ff 20 cc ff a9 04  c9
14c8 : 20 c3 ff 60 00 00 00 00  d6
14d0 : 00 00 00 00 00 00 00 00  d1
14d8 : 00 00 00 00 00 00 00 00  d9
14e0 : 00 00 00 00 00 00 00 00  e1
14e8 : 00 00 00 00 00 00 1b f0  37
```

Listing 3. »Hardmaker 24.obj« erzeugt gestochen scharfe Grafiken auf 24-Nadel-Druckern

ALLES ÜBER GRAFIK

VIC-II-Chip Grafikmöglichkeiten
Darstellung von Zeichen
normaler Zeichen-Modus
Mehrfarben Zeichen-Modus
Modus mit verändertem Hintergrund
eingebauter Zeichensatz
eingebauter Zeichensatz
eingebauter Zeichensatz
programmierbarer Zeichensatz
programmierbarer Zeichensatz
programmierbarer Zeichensatz
Sprites
normale Sprites
mehrfarbige Sprites
Hochauflösungs-modus
normale Hochauflösung
mehrfarbige Hochauflösung

Bild 1. Die Grafikfähigkeiten des VIC-II in der Übersicht

Fortsetzung von Seite 7

nensprache) wird eine Bank durch Steuerung der Bits 0 und 1 von PORT A des CIA # 2 (Adresse 56576 oder $DD00 in Hexadezimaldarstellung) gewählt. Zur Änderung der Speicherabschnitte müssen diese 2 Bit auf Ausgabe gesetzt sein. Dies wird anhand der nachstehenden Beispiele deutlich:

```
POKE 56578, PEEK(56578) OR 3 :REM BITS 0 UND 1 ALS AUSGANG SETZEN
POKE 56576, (PEEK(56576)AND 252)OR A:REM VIDEO-BANK WECHSELN
```

»A« muß einen der in Tabelle 1 dargestellten Werte haben:

Wert von A	Bits	Bank	Start-Platz	Bereich des VIC-II-Chip
0	00	3	49152	($C000-$FFFF)*
1	01	2	32768	($8000-$BFFF)
2	10	1	16384	($4000-$7FFF)*
3	11	0	0	($0000-$3FFF) (Standardwert)

*Anmerkung: Der Zeichensatz des C 64 ist in den Banks 1 und 3 für den VIC-II-Chip nicht verfügbar. (Siehe Abschnitt »Zeichenspeicher«.)

Tabelle 1. Die Bits 0 und 1 von Speicherstelle 56576 regeln den Zugriff des VC-II-Chip auf den Speicher

Dieses Konzept der 16-KByte-Abschnitte spielt bei allen Anwendungen des VIC-II-Chip eine Rolle. Sie sollten stets wissen, auf welche Bank der VIC-II eingestellt ist, da dies beeinflußt, von wo die Zeichendatenmuster kommen, wo sich der Bildschirm befindet, von wo die Sprites kommen usw. Nach dem Einschalten des C 64 sind die Bits 0 und 1 der Speicherstelle 56576 gesetzt, und damit ist Bank 0 ($0000–$3FFF) für sämtliche Anzeigeinformationen eingestellt.

Bildschirmspeicher

Durch POKEn in das Steuerregister 53272 ($D018 HEX) kann die Adresse des Bildschirmspeichers geändert werden. Dieses Register wird jedoch auch zur Steuerung des jeweils benutzten Zeichensatzes verwendet. Achten Sie daher besonders darauf, daß dieser Teil des Steuerregisters nicht verändert wird. Die oberen 4 Bit steuern den Platz des Bildschirmspeichers. Zur Veränderung der Adresse des Bildschirms ist folgende Anweisung erforderlich:

```
POKE 53272,(PEEK(53272)AND 15) OR A
```

Doch diese Anweisung genügt dem C 64 noch nicht. Damit das Betriebssystem weiß, wo der aktuelle Bildschirmspeicher liegt, wird außerdem die Adresse 648 ($0288) abgefragt. Deren Inhalt (PEEK) errechnet sich aus den Adreßwerten der Tabelle 2 nach folgender Formel:

Zahl = Adresse/256

Nach dem Einschalten des C 64 steht in dieser Speicherstelle immer der Standardwert "4", demnach liegt der Bildschirmspeicher im »Normalfall« ab Adresse 1024 (4 x 256 = 1024). Ergänzen Sie also die bereits bekannte Anweisung zum Verschieben des Bildschirms:

```
POKE 53272, (PEEK (53272) AND 15) OR A: POKE 648, Zahl
```

A	BITS	ADRESSE*	
		DEZIMAL	HEXADEZIMAL
0	0000XXXX	0	$0000
16	0001XXXX	1024	$0400 (STANDARD)
32	0010XXXX	2048	$0800
48	0011XXXX	3072	$0C00
64	0100XXXX	4096	$1000
80	0101XXXX	5120	$1400
96	0110XXXX	6144	$1800
112	0111XXXX	7168	$1C00
128	1000XXXX	8192	$2000
144	1001XXXX	9216	$2400
160	1010XXXX	10240	$2800
176	1011XXXX	11264	$2C00
192	1100XXXX	12288	$3000
208	1101XXXX	13312	$3400
224	1110XXXX	14336	$3800
240	1111XXXX	15360	$3C00

*Bitte denken Sie daran, daß die Startadresse der jeweiligen Bank des VIC-II-Chip addiert werden muß.

Tabelle 2. Die möglichen Adressen des Bildschirmspeichers und die entsprechenden Bits in Adresse 53272

Farbspeicher

Der Farbspeicher kann nicht verschoben werden. Er befindet sich stets an den Speicherstellen 55296 ($D800) bis 56295 ($DBE7). Bildschirmspeicher und Farbspeicher werden in den verschiedenen Grafikmodi unterschiedlich benutzt. Ein in einem bestimmten Modus erstelltes Bild sieht in einer anderen Grafikart häufig völlig anders aus.

Zeichenspeicher

Für die Programmierung von Grafiken ist es wesentlich, woher genau der VIC-II die Zeicheninformation bekommt. Normalerweise erhält der Chip die Konturen der anzuzeigenden Zeichen vom Character-Generator-ROM. In diesem Chip werden die Muster gespeichert, die die verschiedenen Buchstaben, Zahlen, Interpunktionssymbole und alle anderen Zeichen der Tastatur bilden. Eines der Merkmale des C64 ist seine Fähigkeit, im RAM-Speicher befindliche Muster zu benutzen. Diese RAM-Muster können von Ihnen beliebig verändert werden, so daß Ihnen ein nahezu unbegrenzter Satz an Symbolen für Spiele, Geschäftsanwendungen etc. zur Verfügung steht.

Ein normaler Zeichensatz enthält 256 Zeichen, bei dem jedes Zeiches durch 8 Byte beschrieben wird. Daher benötigt der komplette Zeichensatz 256 x 8 = 2 KByte. Da der VIC-II-Chip gleichzeitig auf 16 KByte zugreift, gibt es acht mögliche Positionen im Speicher für einen vollständigen Zeichensatz. Sie brauchen natürlich nicht immer einen ganzen Zeichensatz zu verwenden. Er muß jedoch stets an einer der acht möglichen Adressen beginnen. Die Lage des Zeichenspeichers wird durch 3 Bit des VIC-II-Steuerregisters 53272 ($D018 Hex.) kontrolliert. Die Bits 3, 2 und 1 steuern, wo sich der Zeichensatz in 2-KByte-Sätzen befindet. Bit 0 wird überlesen. Bitte denken Sie daran, daß dies das gleiche Register ist, das auch die Lage des Bildschirmspeichers bestimmt. Um die Lage vom Zeichenspeicher zu ändern, benutzen Sie Tabelle 3 und die folgende Basic-Anweisung:

```
POKE 53272,(PEEK(53272) AND 240) OR A
```

WERT VON A	BITS	ADRESSE DES ZEICHENSPEICHERS*	
		DEZIMAL	HEXADEZIMAL
0	XXXX000X	0	$0000-$07FF
2	XXXX001X	2048	$0800-$0FFF
4	XXXX010X	4096	$1000-$17FF In Bank 0 und 2 für Zeichensatz-ROM reserviert.
6	XXXX011X	6144	$1800-$1FFF In Bank 0 und 2 für Zeichensatz-ROM reserviert.
8	XXXX100X	8192	$2000-$27FF
10	XXXX101X	10240	$2800-$2FFF
12	XXXX110X	12288	$3000-$37FF
14	XXXX111X	14336	$3800-$3FFF

*Bitte denken Sie daran, die Startadresse der Bank zu addieren.

Tabelle 3. Die Lage des Zeichenspeichers wird durch Bit 3, 2 und 1 in Adresse 53272 bestimmt

Der VIC-II-Chip kann immer nur auf einen Speicherbereich von 16 KByte zugreifen. Die Muster aller Zeichen sind aber im Zeichen-ROM zwischen den Speicherstellen 53248 und 57343 ($D000 bis $DFFF) bereitgelegt. Sie befinden sich also in der Bank 3. Andererseits ist der VIC-II-Chip nach dem Einschalten des Computers auf die Bank 0 gerichtet, und er sieht deshalb die Bank 3 mit den Zeichenmustern nicht. Die Lösung dieses Problems bilden Spiegelungen aller Zeichenmuster in bestimmte Bereiche der Bank 0 und der Bank 2. Der Videochip geht dann davon aus, daß sich die Zeichenmuster im Bereich 4096 bis 8191 ($1000 bis $1FFF) der Bank 0 und im Bereich 36864 bis 40959 ($9000 bis $9FFF) der Bank 2 befinden, und holt sich von dort die Muster, die er zur Darstellung von Zeichen benötigt.

Damit keine Mißverständnisse entstehen: Rein physikalisch liegen die Zeichenmuster natürlich im Zeichen-ROM ab Adresse $D000, von der Programmlogik aber – die den VIC-II-Chip steuert – her gesehen, existieren die Spiegelbilder. In den meisten Betriebszuständen des Videochip braucht man sich um diese Eigenart der Zeichenspeicherung nicht zu kümmern. Programme und Daten lassen sich ungehindert in den Spiegelbilderbereichen nutzen. Störungen treten erst dann auf, wenn man einen Grafikspeicher in solch eine Spiegelung legt, was man daher möglichst vermeiden sollte. Frei von derartigen Spiegelbildern sind die Banks 1 und 3.

Die Tabelle 4 zeigt Ihnen den Aufbau des Zeichen-ROM und die dazu gehörenden Adressen der Spiegelung in Bank 0.

ADRESSE			VIC-II-SPIEGELUNG	INHALT
BLOCK	DEZIMAL	HEX		
0	53248	D000-D1FF	1000-11FF	Großbuchstaben
	53760	D200-D3FF	1200-13FF	Grafikzeichen
	54272	D400-D5FF	1400-15FF	Großbuchstaben in Reversdarstellung
	54784	D600-D7FF	1600-17FF	Grafikzeichen in Reversdarstellung
1	55296	D800-D9FF	1800-19FF	Kleinbuchstaben
	55808	DA00-DBFF	1A00-1BFF	Großbuchstaben und Grafikzeichen
	56320	DC00-DDFF	1C00-1DFF	Kleinbuchstaben in Reversdarstellung
	56832	DE00-DFFF	1E00-1FFF	Großbuchstaben in Reversdarstellung

Tabelle 4. Anordnung der Zeichenmuster im Zeichen-ROM und der Spiegelung in Bank 0

Dem aufmerksamen Leser wird jetzt aufgefallen sein, daß die vom Zeichen-ROM beanspruchten Adressen die gleichen sind, wie die der VIC-II-Chip-Steuerregister. Dies ist möglich, da diese Adressen nicht gleichzeitig beansprucht werden.

Benötigt der VIC-II-Chip den Zugriff auf die Zeichendaten, so wird das ROM eingeschaltet. In der 16-KByte-Speicherbank, auf die der VIC-II-Chip zugreift, entsteht die entsprechende »Spiegelung«. Ansonsten wird dieser Bereich von den Ein-/Ausgaberegistern beansprucht und das Zeichen-ROM kann nur vom VIC-II erreicht werden.

Es kann jedoch passieren, daß Sie das Zeichen-ROM benötigen, und zwar dann, wenn Sie programmierbare Zeichen benutzen wollen und eine Kopie eines Teils vom Zeichen-ROM für die Zeichendefinition benötigen. In diesem Fall müssen Sie das Ein-/Ausgaberegister aus- und das Zeichen-ROM einschalten. Dann können Sie kopieren. Danach muß das Ein-/Ausgaberegister erneut eingeschaltet werden. Während des Kopierens (bei ausgeschalteter Ein-/Ausgabe) sind keine Unterbrechungen erlaubt. Für Unterbrechungen werden nämlich die Ein-/Ausgaberegister benötigt. Wenn Sie dies vergessen und eine Unterbrechung vornehmen, passiert Unvorhersehbares. Die Tastaturabfrage darf während des Kopierens nicht vorgenommen werden. Um die Tastatur und weitere normale Unterbrechungen abzuschalten, die mit dem C64 möglich sind, benutzen Sie folgende POKE-Anweisung im Programm:

```
POKE 56334, PEEK(56334) AND 254: REM
(Interrupt AUS)
```

Wenn Sie den Zugriff auf das Zeichen-ROM beendet haben und bereit sind für die Programmfortsetzung, wird die Tastatur durch folgende POKE-Anweisung wieder eingeschaltet:

```
POKE 56334, PEEK(56334) OR1: REM (Interrupt EIN)
```

Durch folgende POKE-Anweisung wird die Ein-/Ausgabe

ausgeschaltet und das Zeichen-ROM eingeschaltet:

```
POKE 1, PEEK(1) AND 251
```

Das Zeichen-ROM befindet sich nun an den Speicherstellen 53248 bis 57343 ($D000–$DFFF). Um die Ein-/Ausgabe für den normalen Betrieb zurück in $D000 zu schalten, benutzen Sie folgende POKE-Anweisung:

```
POKE 1, PEEK(1) OR 4
```

Programmierbare Zeichen

Beim Einschalten des C64 befindet sich dieser im Standardzeichenmodus. Dies ist der Modus, in dem Sie normalerweise Programmierungen vornehmen.

Zeichen können aus dem ROM oder dem RAM gelesen werden. Normalerweise wird jedoch auf die Zeichen im ROM zugegriffen. Benötigen Sie für ein Programm spezielle Grafikzeichen, so brauchen Sie lediglich die neuen Zeichenmuster im RAM zu definieren und den VIC-II-Chip anzuweisen, die Zeicheninformationen von da und nicht aus dem Zeichen-ROM zu nehmen. Dies wird im nachstehenden Abschnitt noch genauer beschrieben.

Um Zeichen auf dem Bildschirm in Farbe anzuzeigen, greift der VIC-II-Chip auf den Bildschirmspeicher zu, um den Zeichen-Code für diese Bildschirmposition zu bestimmen. Gleichzeitig greift er auf den Farbspeicher zu, um die Farbe des Zeichens festzulegen. Der Zeichen-Code wird vom VIC-II in die Startadresse des 8-Byte-Satzes mit Ihrem Zeichenmuster umgesetzt. Dieser Satz befindet sich im Zeichenspeicher.

Die Umsetzung ist nicht zu kompliziert, zur Erstellung der gewünschten Adresse werden jedoch verschiedene Punkte kombiniert. Zunächst wird der von Ihnen bei der POKE-Anweisung für den Bildschirmspeicher benutzte Zeichencode mit 8 multipliziert. Danach wird der Anfang vom Zeichenspeicher addiert (siehe Abschnitt »Zeichenspeicher«). Nun werden die Bankanwahl-Bits berücksichtigt. Hierzu wird die Basisadresse (siehe Abschnitt »Video-Bankanwahl«) addiert. Anhand der folgenden einfachen Gleichung können Sie sehen, wie dies gemeint ist:

Zeichenadresse = Bildschirmcode *8 + (Zeichensatz *1024)+(Bank*16384).

Zeichendefinitionen

Jedes Zeichen wird aus einer Matrix von 8 x 8 Punkten gebildet. Hierbei können die einzelnen Punkte entweder ein- oder ausgeschaltet sein. Beim C64 sind die Zeichenbilder im Zeichengenerator-ROM abgelegt. Jedes Zeichen ist hierbei als Satz von 8 Byte gespeichert. Jedes Byte steht für das Punktmuster einer Reihe im Zeichen und jedes Bit für einen Punkt. Ein 0-Bit zeigt an, daß der Punkt ausgeschaltet und ein 1-Bit, daß er eingeschaltet ist.

Der Zeichenspeicher im ROM beginnt bei Adresse 53248 (bei ausgeschalteter Ein-/Ausgabe). Die ersten 8 Byte von Adresse 53248 ($D000) bis 53255 ($D007) enthalten das Muster für das Zeichen »@«, dessen Zeichencodewert im Bildschirmspeicher 0 ist.

Die nächsten 8 Byte von Adresse 53256 ($D008) bis 53263 ($D00F) enthalten die Information zur Bildung des Buchstabens A.

BELEGUNG	BINÄR	PEEK
**	00011000	24
****	00111100	60
** **	01100110	102
******	01111110	126
** **	01100110	102
** **	01100110	102
** **	01100110	102
	00000000	0

Jeder vollständige Zeichensatz beansprucht eine Speicherkapazität von 2 KByte (2048 Byte). Insgesamt sind 256 Zeichen enthalten, wobei jedes Zeichen 8 Byte umfaßt. Da es insgesamt zwei Zeichensätze gibt, und zwar einen für die Großbuchstaben und Grafikzeichen und den anderen für Groß- und Kleinbuchstaben, belegt der ROM-Zeichenspeicher insgesamt 4 KByte Speicherplatz.

Programmierbare Zeichen

Da die Zeichen im ROM gespeichert sind, sieht es so aus, als ob sie für frei programmierbare Zeichen nicht geändert werden könnten. Der Speicherplatz, der dem VIC-II-Chip mitteilt, wo die Zeichen zu finden sind, ist jedoch ein programmierbares Register. Dieses kann so geändert werden, daß es auf viele Spielbereiche zeigt. Indem der Zeichenspeicherzeiger so geändert wird, daß er auf das RAM zeigt, kann der Zeichensatz beliebig programmiert werden.

Soll sich Ihr Zeichensatz im RAM befinden, so gibt es einige sehr wichtige Regeln, die Sie dabei berücksichtigen müssen.

1) Dies ist ein Alles- oder Nichts-Vorgang. Im allgemeinen, wenn Sie den VIC-II-Chip angewiesen haben, die Zeicheninformation aus dem vorbereiteten RAM-Bereich zu nehmen, sind die Standardzeichen vom C64 für Sie nicht verfügbar. Um dieses Problem zu lösen, müssen Sie alle Buchstaben, Zahlen oder Standardgrafikzeichen vom C64 in den RAM-Speicher kopieren, den Sie dann in Ihrem Programm benutzen wollen. Hierbei können Sie beliebige Zahlen auswählen und brauchen auch nicht auf die Reihenfolge zu achten!
2) Ihr Zeichensatz benutzt denselben Speicher wie Basic-Programme. Da hierfür jedoch 38 KByte zur Verfügung stehen, ist dies meist problemlos.

Achten Sie jedoch darauf, daß Ihr Zeichensatz nicht von Variablen, die das Basic-Programm anlegt, überschrieben wird.

Zwei Adressen in der Bank 0 des C64 dürfen nicht als Beginn des Zeichensatzes gewählt werden: Adresse 0 und Adresse 2048. Die erste darf nicht benutzt werden, da das System auf Seite 0 (Zero-Page) wichtige Daten speichert und Adresse 2048 ist der Beginn Ihres Basic-Programms!

Für Ihren Zeichensatz stehen jedoch noch sechs weitere Anfangspositionen zur Verfügung.

Am besten wählen Sie hierzu am Anfang Adresse 12288 ($3000 in Hexadezimaldarstellung). Dies erfolgt durch die nun folgende POKE-Anweisung:

```
POKE 53272,(PEEK(53272) AND 240)OR 12
```

Sofort sind alle Buchstaben vom Bildschirm verschwunden. Der Grund hierfür liegt darin, daß bis jetzt noch kein Zeichensatz ab Adresse 12288 steht, sondern nur zufällige Bytes. Kehren Sie mit dem C64 durch Betätigung der Tasten <RUN/STOP> und <RESTORE> wieder zurück in den Normalmodus.

Nun wollen wir Grafikzeichen erstellen. Um Ihren Zeichensatz zu schützen, sollten Sie die Speicherkapazität für Basic-Programme reduzieren. Der Speicher in Ihrem Computer bleibt unverändert. Sie haben dem C 64 lediglich die Anweisung gegeben, einen bestimmten Teil des Basic-Speichers nicht zu benutzen. Tippen Sie folgendes ein:

```
PRINT FRE(0)-(SGN(FRE(0))<0)*65535
```

Die angezeigte Zahl gibt die unbenutzte Speicherkapazität an. Geben Sie nun folgendes ein:

```
POKE 52,48:POKE56,48:CLR
```

Und nun:

```
PRINT FRE(0)-(SGN(FRE(0))<0)*65535
```

Sehen Sie die Änderung?

Der C64 nimmt nun an, daß weniger Speicherkapazität zur Verfügung steht. In diesen gewonnenen Speicherplatz können Sie nun Ihren Zeichensatz eingeben.

```
5 PRINT CHR$(142)                    :REM GR
  OSSBUCHSTABEN EINSCHALTEN                <060>
10 POKE 52,48:POKE 56,48:CLR         :REM BAS
   ICSPEICHER BEGRENZEN                    <059>
20 POKE 56334,PEEK(56334) AND 254 :REM UNT
   ERBRECHUNGEN ABSCHALTEN                 <061>
30 POKE 1,PEEK(1) AND 251            :REM ZEI
   CHENROM EINBLENDEN                      <060>
40 FOR I=0 TO 511:POKE I+12288,PEEK(I+5324
   8):NEXT                                 <039>
50 POKE 1,PEEK(1) OR 4               :REM EIN
   -/AUSGABE EINBLENDEN                    <115>
60 POKE 56334,PEEK(56334) OR 1       :REM UNT
   ERBRECHUNGEN ZULASSEN                   <250>
70 END                                     <072>
```

Listing 1. 64 Zeichen werden aus dem RAM ins ROM übertragen

Als nächstes müssen nun Ihre Zeichen ins RAM kopiert werden. Durch ein kurzes Programm werden 64 Zeichen vom ROM in den RAM-Zeichensatz übertragen (Listing 1).

Geben Sie nun ein:

```
POKE 53272,(PEEK(53272)AND 240) OR 12
```

Nichts passiert, stimmt's? Fast nichts! Der C64 bekommt die Zeicheninformationen nun vom RAM und nicht vom ROM. Da wir jedoch die Zeichen genau vom ROM kopiert haben, ist kein Unterschied zu sehen...noch nicht.

Die Zeichen können nun leicht geändert werden. Löschen Sie den Bildschirm, und drücken Sie die Taste <@>. Bewegen Sie den Cursor um einige Zeilen nach unten, und geben Sie dann folgendes ein:

```
FOR I = 12288 TO 12288 +7:POKE I,255 - PEEK(I) :
NEXT
```

Sie haben soeben durch Umkehrung der Bitmuster im Zeichenspeicher ein <@> in Reversdarstellung erstellt!

Bewegen Sie nun den Cursor wieder zum Programmanfang und drücken Sie <RETURN> erneut, um das Zeichen noch einmal umzukehren (das heißt, es wird wieder normal dargestellt).

Denken Sie daran, daß zur Speicherung jedes Zeichens 8 Byte benötigt werden. Mit der Bildschirmcode-Tabelle in Ihrem C64-Handbuch können Sie die Position des Zeichens im RAM ermitteln. In Tabelle 5 finden Sie einige Beispiele.

ZEICHEN	BILDSCHIRM-CODE	DERZEITIGE STARTADRESSE IM RAM
@	0	12288
A	1	12296
!	33	12552
>	62	12784

Tabelle 5. Beispiele für die Startadressen der Bit-Muster einzelner Zeichen im RAM

Wir haben nur die ersten 64 Zeichen übernommen. Wird eines der anderen Zeichen gewünscht, so ist vorher noch etwas zu berücksichtigen. Was ist zu tun, wenn Sie nun Zeichennummer 154, ein umgekehrtes »Z« wünschen? Sie können das erreichen, indem Sie ein »Z« - wie gezeigt - umkehren, oder Sie können den Satz der umgekehrten Zeichen vom ROM kopieren oder einfach das eine Zeichen aus dem ROM holen und ein nicht benötigtes Zeichen im RAM dadurch ersetzen.

Nehmen wir an, Sie benötigen das Zeichen »>« nicht mehr. Dieses Zeichen soll also gegen das negativ dargestellte »Z« ausgetauscht werden. Geben Sie folgendes ein:

```
FOR I = 0 TO 7:POKE 12784 + I,255-PEEK(I+12496):
NEXT
```

Geben Sie nun »>« ein. Es erscheint als umgekehrtes »Z«. So oft Sie nun dieses auch eingeben, erscheint es immer als umgekehrtes »Z«. (Diese Änderung betrifft jedoch

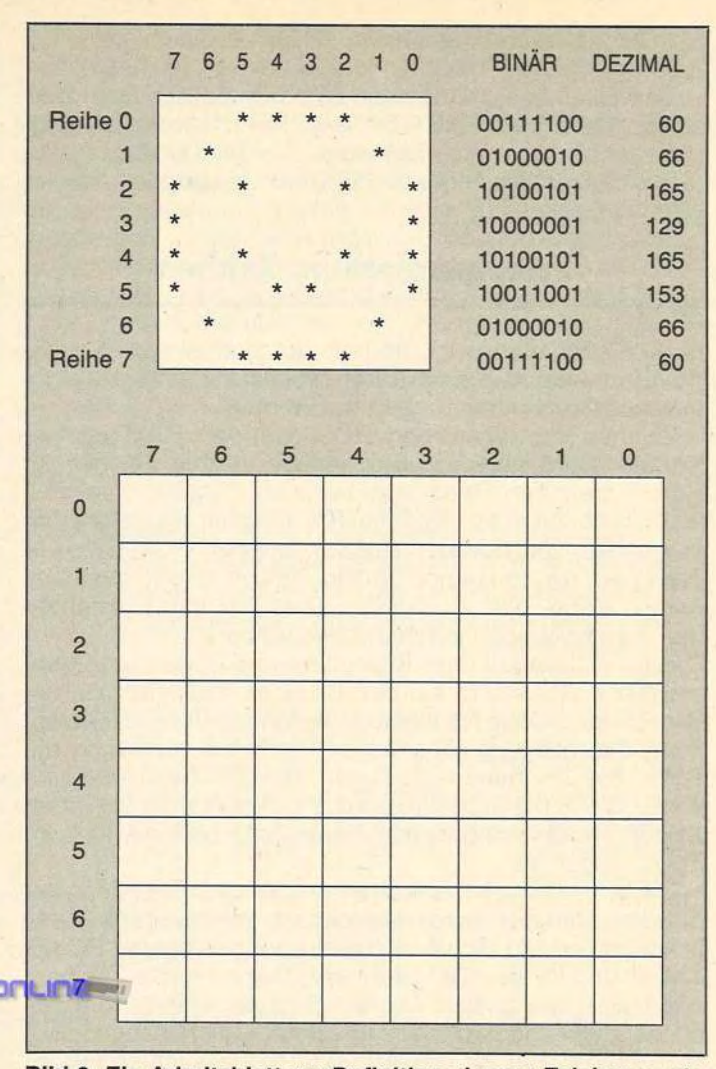


Bild 2. Ein Arbeitsblatt zur Definition eigener Zeichenmuster

nur die Darstellung auf dem Bildschirm. Auch wenn das Zeichen wie ein umgekehrtes »Z« aussieht, wirkt es in einem Programm doch immer noch als »>«.) Probieren Sie das an einem Beispiel aus, bei dem dieses Zeichen benötigt wird.

Fassen wir zusammen: Sie können nun Zeichen vom ROM in das RAM kopieren. Sie können hierbei selbst die Zeichen auswählen. Hinsichtlich der programmierbaren Zeichen fehlt Ihnen also nur noch ein Punkt (und zwar der beste!)....das Erstellen Ihrer eigenen Zeichen.

Jedes Zeichen wird als Gruppe von 8 Byte gespeichert. Die Bit-Muster der Bytes geben direkt das Zeichen wieder. Werden 8 Byte übereinander angeordnet und jedes Byte als achtstellige Binärzahl geschrieben, so entsteht eine 8 x 8-Matrix. Ist ein Bit eine 1, so ist an diesem Platz ein Punkt. Ist ein Bit eine 0, ist an diesem Platz eine Leerstelle.

Zum Erstellen Ihrer eigenen Zeichen geben Sie in den Speicher die entsprechende Bit-Anordnung ein. Geben Sie NEW und danach dieses Programm ein:

```
10 FOR I = 12448 TO 12455 : READ A: POKE I,A : NEXT
20 DATA 60, 66, 165, 129, 165, 153, 66, 60
```

Geben Sie nun RUN ein. Das Programm ersetzt den Buchstaben T durch ein »Gesicht«. Um das Gesicht zu sehen, geben Sie mehrere Ts ein. Jede Zahl in der DATA-Anweisung in Zeile 20 ist eine Reihe in diesem Gesicht. Es gilt die Matrix in Bild 2.

Das Schema in Bild 2 hilft Ihnen beim Entwurf Ihrer eigenen Zeichen. Das Blatt enthält eine Matrix von 8 x 8 mit Reihennummern sowie Nummern über jeder Spalte. (Wird jede Reihe als Binärwert gesehen, so sind die Spaltennummern die jeweiligen Werte der Bit-Position. Der Wert läßt

sich einfach als Zweierpotenz errechnen. Das linke äußerste Bit entspricht 128 oder 2^7, das nächste 64 oder 2^6 usw., bis das äußerste rechte Bit (Bit 0) erreicht ist. Bit 0 entspricht 1 oder 2^0.)

Tragen Sie in die Matrix überall da ein »X« ein, wo in Ihrem Zeichen ein Punkt erscheinen soll. Ist das Zeichen fertig, dann können Sie die DATA-Anweisung dafür errechnen.

Beginnen Sie mit der ersten Reihe. Überall da, wo ein »X« eingesetzt ist, lesen Sie die Nummer oben an der Spalte ab (die Zweierpotenz), und notieren Sie sie. Dann werden die Zweierpotenzen der ersten Reihe addiert. Notieren Sie diese Summe neben der Reihe. Sie wird später in der DATA-Anweisung benutzt, um diese Reihe als Bitmuster wiederzugeben.

Das gleiche gilt für die übrigen Reihen (1 bis 7). Sie müssen insgesamt acht Zahlen zwischen 0 und 255 haben. Liegt eine dieser Zahlen nicht innerhalb dieses Bereiches, überprüfen Sie die Addition. Bei richtiger Addition müssen die Zahlen auf jeden Fall in diesem Bereich liegen. Haben Sie weniger als acht Zahlen, dann haben Sie wahrscheinlich eine Reihe vergessen. Es ist durchaus korrekt, wenn auch Nullen dabei sind. Diese Null-Reihen sind genauso wichtig, wie die anderen Zahlen.

Ersetzen Sie die Zahlen in der DATA-Anweisung in Zeile 20 durch die soeben berechneten Zahlen und geben Sie danach RUN ein. Drücken Sie nun die Taste <T>. Bei jedem Betätigen dieser Taste sehen Sie Ihr eigenes Zeichen.

lösungsmodus können alle Punkte innerhalb der 8 x 8-Matrix entweder in der Vordergrund- oder in der Hintergrundfarbe angezeigt werden. Hierdurch wird die Farbauflösung innerhalb dieses Bereiches eingeschränkt. Ein Zeichen ist daher nur in einer Farbe darstellbar.

Das Problem läßt sich durch den Mehrfarbenmodus des VIC-II-Chip lösen. Hierbei kann jeder Punkt eine von vier Farben annehmen: Bildschirmfarbe (Hintergrundfarbregister 0), die Farbe im Hintergrundregister 1, die Farbe im Hintergrundregister 2 oder die Zeichenfarbe. Es gibt allerdings eine Einschränkung: Im Mehrfarbenmodus ist jeder Punkt doppelt so breit wie im Hochauflösungsmodus, weshalb die horizontale Auflösung nur noch die Hälfte beträgt. Befindet sich der VIC-II-Chip in diesem Modus, dann lassen sich auch Zeichen mehrfarbig darstellen.

Zum Einschalten des Modus für mehrfarbige Zeichen wird das Bit 4 der Speicherstelle 53270 ($D016) durch folgende POKE-Anweisung auf 1 gesetzt:

```
POKE 53270,PEEK(53270) OR 16
```

Zum Abschalten dieser Betriebsart wird Bit 4 an Speicherplatz 53270 durch nachstehende POKE-Anweisung wieder gelöscht:

```
POKE 53270,PEEK(53270) AND 239
```

Der Mehrfarben-Modus wird für jede Bildschirmposition ein- oder ausgeschaltet, so daß Mehrfarbengrafiken und

```
10 REM * BEISPIEL 1 *                          <017>
20 REM ERZEUGEN EIGENER ZEICHEN                <137>
31 POKE 56334,PEEK(56334) AND 254:POKE 1,P
   EEK(1) AND 251                              <054>
35 FOR I=0 TO 63               :REM ZU KOP
   IERENDER ZEICHENBEREICH                     <012>
36 FOR J=0 TO 7                :REM ALLE 8
    BYTES PRO ZEICHEN                          <014>
37 POKE 12288+I*8+J,PEEK(53248+I*8+J):REM
   EIN BYTE KOPIEREN                           <110>
38 NEXT J:NEXT I               :REM NFCHST
   ES BYTE ODER ZEICHEN                        <164>
39 POKE 1,PEEK(1) OR 4:POKE 56334,PEEK(563
   34) OR 1                                    <192>
40 POKE 53272,(PEEK(53272) AND 240) + 12:R
   EM ZEICHEN AUS RAM HOLEN                    <203>
60 FOR ZEIC=60 TO 63           :REM NEUE Z
   EICHEN 60 BIS 63                            <007>
80 FOR BYTE=0 TO 7             :REM ALLE 8
    BYTES PRO ZEICHEN                          <162>
100 READ NUMBER                :REM EINLES
    EN 1/8 ZEICHEN                             <050>
120 POKE 12288+(8*ZEIC)+BYTE,NUMBER:REM DA
    TEN SPEICHERN                              <118>
140 NEXT BYTE:NEXT ZEIC                        <254>
150 PRINT CHR$(147)TAB(255)CHR$(60);           <251>
155 PRINT CHR$(61)TAB(55)CHR$(62)CHR$(63)      <025>
160 REM BRINGT DIE NEUDEFINIERTEN ZEICHEN
    AUF BILDSCHIRM                             <019>
170 GET A$                     :REM ERWART
    ET TASTENDRUCK                             <240>
180 IF A$="" THEN GOTO 170     :REM NEUER
    VERSUCH WENN KEINE TASTE                   <125>
190 POKE 53272,21              :REM ZURUEC
    K ZU NORMALEN ZEICHEN                      <127>
200 DATA 4,6,7,5,7,7,3,3       :REM DATEN
    FUER ZEICHEN 60                            <138>
210 DATA 32,96,224,160,224,224,192,192:REM
     ZEICHEN 61                                <035>
220 DATA 7,7,7,31,31,95,143,127:REM ZEICHE
    N 62                                       <045>
230 DATA 224,224,224,248,248,248,240,224:R
    EM ZEICHEN 63                              <204>
240 END                                        <242>
```

Listing 2. Ein nettes Beispiel für selbstdefinierte Zeichen

Dies funktioniert jedoch nur, solange die Bereichsumschaltung für das Zeichensatz-ROM noch aktiv ist. <RUN/STOP RESTORE> schaltet nämlich, wie auch die entsprechenden POKEs in Adresse 53272, wieder auf die Standard-Speicheraufteilung um.

Wenn Ihnen dieses Zeichen noch nicht gefällt, ändern Sie einfach die Zahlen in der DATA-Anweisung, bis die Zeichendarstellung zufriedenstellend ist. Das ist alles!

Mit unserem Beispielprogramm (Listing 2) lassen sich Zeichen schon recht komfortabel ändern.

Der Multicolor-Modus

Die bisher gezeigten Zeichen wurden im sogenannten Hochauflösungsmodus erstellt. Das bedeutet, daß für jeden Punkt im Zeichenspeicher zwei Werte zur Verfügung stehen: 1 für EIN und 0 für AUS. Hat ein Punkt den Wert 1, dann wird er in der von Ihnen für die jeweilige Bildschirmoperation gewählten Farbe angezeigt. In diesem Hochauf-

Grafiken mit hoher Auflösung (Hires) kombiniert werden können. Dies läßt sich über Bit 3 im Farbspeicher steuern. Der Farbspeicher beginnt bei 55296 ($D800 HEX). Ist die Zahl im Farbspeicher kleiner als 8 (0 bis 7), so gilt für die entsprechende Stelle auf dem Bildschirm Hochauflösung in der gewählten Farbe (0 bis 7). Ist die Zahl größer oder gleich 8 (von 8 bis 15), dann wird die entsprechende Stelle im Mehrfarbenmodus angezeigt.

Die Zeichenfarbe an einer Bildschirmposition kann durch eine POKE-Anweisung im Farbspeicher geändert werden. Durch das POKEn einer Zahl von 0 bis 7 werden die Zeichen in normaler Farbdarstellung angezeigt. Durch das POKEn einer Zahl zwischen 8 und 15 gilt für die entsprechende Bildschirmstelle der Mehrfarbenmodus, durch das Einschalten von Bit 3 im Farbspeicher wird also der Multicolor-Modus und durch Ausschalten der normale Hochauflösungsmodus gewählt.

Gilt für eine Bildschirmstelle der Mehrfarbenbetrieb, so wird durch Zeichen-Bits bestimmt, welche Farben für die Punkte angezeigt werden. In der folgenden Tabelle sehen

Sie beispielsweise die Darstellung des Buchstabens A und das entsprechende Bit-Muster:

```
BIT-MUSTER
   **      00011000
  *****    00111100
 **  **    01100110
 ******    01111110
 **  **    01100110
 **  **    01100110
 **  **    01100110
           00000000
```

Im normalen oder »Hires«-(das heißt hochauflösenden Grafik-)Modus wird die Bildschirmfarbe bei jedem 0-Bit und die Zeichenfarbe stets da angezeigt, wo das Bit 1 ist. Beim Mehrfarbenmodus werden die Bits auf folgende Art paarweise benutzt:

```
BIT-MUSTER
  AABB      00 01 10 00
  CCCC      00 11 11 00
AABBAABB    01 10 01 10
AACCCCBB    01 11 11 10
AABBAABB    01 10 01 10
AABBAABB    01 10 01 10
AABBAABB    01 10 01 10
            00 00 00 00
```

Im obigen Bildbereich werden die durch AA gekennzeichneten Stellen in der Hintergrundfarbe #1, die durch BB gekennzeichneten Stellen in der Hintergrundfarbe #2 und die durch CC gekennzeichneten Stellen in der Zeichenfarbe dargestellt. Dies zeigt Ihnen die Tabelle 6.

BIT-PAAR	FARBREGISTER	SPEICHERPLATZ
00	Hintergrundfarbe #0 (Bildschirmfarbe)	53281 ($D021)
01	Hintergrundfarbe #1	53282 ($D022)
10	Hintergrundfarbe #2	53283 ($D023)
11	Durch die unteren 3 Bit im Farbspeicher bestimmte Farbe	Farbspeicher

Tabelle 6. Zusammenhang zwischen dem Wert eines Bitpaares und der Herkunft der Farbe des dazugehörenden Punktes

Geben Sie das Listing 3 ein und starten es mit RUN:

```
100 POKE 53281,1                :REM HINTE
    RGRUNDFARBE 0 AUF WEISS                 <096>
110 POKE 53282,3                :REM HINTE
    RGRUNDFARBE 1 AUF CYAN                  <150>
120 POKE 53283,8                :REM HINTE
    RGRUNDFARBE 2 AUF ORANGE                <084>
130 POKE 53270,PEEK(53270) OR 16:REM MULTI
    COLORMODUS EINSCHALTEN                  <065>
140 C=13*4096+8*256             :REM START
    ADRESSE FARBSPEICHER = C                <085>
150 PRINT CHR$(147)"AAAAAAAAAA"             <038>
160 FOR L=0 TO 9                            <254>
170 POKE C+L,8                  :REM BENUT
    ZT WIRD MULTICOLOR SCHWARZ              <250>
180 NEXT                                    <190>
```

Listing 3.Ein Programm zum Erzeugen selbstdefinierter Multicolor-Zeichen

Die Bildschirmfarbe ist weiß, die Zeichenfarbe schwarz, ein Farbregister zyan (grünblau) und das andere orange.

Sie geben nicht tatsächlich Farb-Codes in die Speicher für die Zeichenfarbe ein, sondern benutzen eigentlich Hinweise auf die jeweiligen Farbregister. Hierdurch wird Speicherplatz gespart, da zwei Bits benutzt werden, um zwischen 16 beziehungsweise 8 Farben (Hintergrund beziehungsweise Zeichen) zu wählen. Hierdurch werden einige raffinierte Tricks möglich. Durch einfaches Ändern eines

```
100 POKE 53270,PEEK(53270)  OR 16           <032>
110 PRINT CHR$(147)CHR$(18);                <185>
120 PRINT"{ORANGE}";                        <025>
130 FOR L=1 TO 22:PRINT CHR$(65);:NEXT      <169>
135 FOR T=1 TO 500:NEXT                     <207>
140 PRINT"{BLUE}";                          <006>
145 FOR T=1 TO 500:NEXT                     <217>
150 PRINT"{BLACK}BITTE EINE TASTE"          <121>
160 GET A$:IF A$="" THEN 160                <162>
170 X=INT(RND(1)*16)                        <072>
180 POKE 53282,X                            <055>
190 GOTO 160                                <214>
```

Listing 4. Mit <SPACE> werden in diesem Demo die Farben im Multicolor-Modus verändert

```
10 REM * BEISPIEL 2 *                       <033>
20 REM ERZEUGEN MEHRFARBIGER SELBSTDEFINIE
   RTER ZEICHEN                             <000>
31 POKE 56334,PEEK(56334) AND 254:POKE 1,P
   EEK(1) AND 251                           <054>
35 FOR I=0 TO 63           :REM ZU KOPIEREN
   DE ZEICHEN                               <203>
36 FOR J=0 TO 7            :REM ALLE 8 BYTE
   S PRO ZEICHEN                            <014>
37 POKE 12288+I*8+J,PEEK(53248+I*8+J):REM
   KOPIERE EIN BYTE                         <218>
38 NEXT J,I                :REM NAECHSTES B
   YTE ODER ZEICHEN                         <078>
39 POKE 1,PEEK(1) OR 4:POKE 56334,PEEK(563
   34) OR 1                                 <192>
40 POKE 53272,(PEEK(53272) AND 240) + 12:R
   EM ZEICHENRAM EINSCHALTEN                <094>
50 POKE 53270,PEEK(53270) OR 16             <238>
51 POKE 53281,0            :REM HINTERGRUND
   FARBE 0 AUF SCHWARZ                      <000>
52 POKE 53282,2            :REM HINTERGRUND
   FARBE 1 AUF ROT                          <047>
53 POKE 53283,7            :REM HINTERGRUND
   FARBE 2 AUF GELB                         <157>
60 FOR ZEIC=60 TO 63       :REM NEUE ZEICHE
   N 60 BIS 63                              <007>
80 FOR BYTE=0 TO 7         :REM ALLE 8 BYTE
   S PRO ZEICHEN                            <162>
100 READ ZAHL              :REM LESE 1/8 ZE
    ICHENDATEN                              <076>
120 POKE 12288+(8*ZEIC)+BYTE,ZAHL:REM SPEI
    CHERN IM ZEICHENRAM                     <184>
140 NEXT BYTE,ZEIC                          <081>
150 PRINT"{CLR}"TAB(255)CHR$(60)CHR$(61)TA
    B(55)CHR$(62)CHR$(63)                   <239>
160 REM ZEILE 150 BRINGT DIE NEUEN ZEICHEN
    AUF DEN BILDSCHIRM                      <048>
170 GET A$                 :REM ERWARTET TA
    STENDRUCK                               <240>
180 IF A$="" THEN 170      :REM SONST NEUER
    VERSUCH                                 <083>
190 POKE 53272,21:POKE 53270,PEEK(53270) A
    ND 239:REM NORMALER ZEICHENSATZ         <032>
200 DATA 129,37,21,29,93,85,85,85 :REM ZEI
    CHENDATEN FUER ZEICHEN 60               <160>
210 DATA 66,72,84,116,117,85,85,85:REM ZEI
    CHEN 61                                 <026>
220 DATA 87,87,85,21,8,8,40,0      :REM ZEI
    CHEN 62                                 <028>
230 DATA 213,213,85,84,32,32,40,0 :REM ZEI
    CHEN 63                                 <247>
240 END                                     <242>
```

Listing 5. Ein programmierbares Multicolor-Zeichen

der indirekten Register wird jeder Punkt, der in dieser Farbe gezeichnet ist, ebenfalls geändert.

Alles, was in Bildschirm- und Hintergrundfarben angezeigt ist, kann daher sofort auf dem gesamten Bildschirm geändert werden. In Listing 4 sehen Sie ein Beispiel zur Änderung des Hintergrundfarbregisters #1.

Über die Taste <CBM> und die Farbtasten können allen Zeichen, einschließlich den Mehrfarbenzeichen, beliebige Farben gegeben werden. Geben Sie zum Beispiel folgende Befehle ein:

```
POKE 53270,PEEK(53270)OR 16:PRINT"{CTRL3}";:
```

Das Wort READY und alles übrige, was Sie über die Tastatur eingeben, wird im Mehrfarbenmodus angezeigt. Durch eine andere Farbsteuerung können Sie wieder den Normal-Modus wählen. Listing 5. zeigt Ihnen ein Programmbeispiel mit programmierbaren Mehrfarbenzeichen.

Erweiterter Hintergrundfarbmodus

Im erweiterten Hintergrundfarbmodus können Sie für jedes einzelne Zeichen die Farbe sowohl im Hintergrund als auch im Vordergrund steuern. So ist es zum Beispiel möglich, auf einem weißen Bildschirm ein blaues Zeichen mit gelbem Hintergrund anzuzeigen.

Für diesen Betriebszustand des VIC-II-Chip stehen vier Register zur Verfügung. Für jedes Register kann eine der 16 Farben gewählt werden.

Die Vordergrundfarbe wird wie im normalen Textmodus über den Farbspeicher festgelegt.

Beim erweiterten Modus ist die Anzahl der verschiedenen anzeigbaren Zeichen jedoch eingeschränkt. Ist der erweiterte Farbmodus eingeschaltet, können nur die ersten 64 Zeichen des Zeichen-ROM (oder die ersten 64 in Ihrem programmierbaren Zeichensatz) benutzt werden. Zwei Bits des Zeichen-Codes werden nämlich für die Wahl der Hintergrundfarbe benutzt.

Der Zeichen-Code (die auf dem Bildschirm gePOKEte Zahl) vom Buchstaben »A« ist eine 1. Im erweiterten Farbmodus erscheint nach dem POKEn einer 1 ein »A«. Normalerweise müßte nach dem POKEn von 65 das Zeichen mit dem Zeichen-Code (CHR$) 129, also ein reverses »A« erscheinen. Dies passiert nicht im erweiterten Farbmodus. Es erscheint genau das gleiche »A« wie vorher, jedoch vor einer anderen Hintergrundfarbe. Entnehmen Sie die Codes Tabelle 7:

ZEICHENCODE			HINTERGRUNDFARBREGISTER	
BEREICH	BIT 7	BIT6	NUMMER	ADRESSE
0- 63	0	0	0	53281 ($D021)
64-127	0	1	1	53282 ($D022)
128-191	1	0	2	53283 ($D023)
192-255	1	1	3	53284 ($D024)

Tabelle 7. Zusammenhang der Werte in Bit 6 und 7 des Zeichencodes mit den Hintergrundfarbregistern im erweiterten Hintergrundmodus

Zum Einschalten des erweiterten Farbmodus wird Bit 6 des VIC-II-Registers mit der Adresse 53265 ($D011 in HEX) auf 1 gesetzt. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53265, PEEK(53265) OR 64
```

Zum Ausschalten des erweiterten Farbmodus löscht man dieses Bit. Hierzu dient folgende Anweisung:

```
POKE 53265, PEEK (53265) AND 191
```

Grafiken durch Bit-Mapping

Beim Programmieren von Spielen, Zeichnen von Tabellen für Geschäftsanwendungen oder Schreiben von sonstigen Programmen, werden Sie früher oder später Bildschirmdarstellungen mit hoher Auflösung benötigen.

Der C64 wurde genau hierfür konstruiert: Hohe Auflösung wird durch »Bit-Mapping« des Bildschirms möglich. »Bit-Mapping« ist die Methode, bei der jedem darstellbaren Punkt (Pixel) auf dem Bildschirm sein eigenes Bit (Platz) im Speicher zugeordnet wird. Ist dieses Speicherbit eine 1, so ist der entsprechende Punkt eingeschaltet. Ist das Bit 0, so ist der Punkt ausgeschaltet.

Das Arbeiten mit Grafiken bei hoher Auflösung hat jedoch einige Nachteile und wird daher nicht immer benutzt. Zunächst wird durch das Bit-Mapping des gesamten Bildschirms eine erhebliche Speicherkapazität in Anspruch genommen. Jeder Pixel benötigt nämlich ein Speicherbit, das heißt, Sie brauchen 1 Byte für 8 Pixel. Da jedes Zeichen eine 8 x 8-Matrix ist und 40 Zeilen mit 25 Zeichen vorhanden sind, beträgt die Auflösung 320 Pixel (Punkte) x 200 Pixel für den gesamten Bildschirm. Hieraus ergeben sich 64000 Punkte, von denen jeder ein Speicherbit benötigt. Für ein Bit-Mapping des gesamten Bildschirms brauchen Sie also 8000 Byte.

Bit-Mapping ist die am weitesten verbreitete Grafiktechnik in der Computerwelt. Dieses Verfahren wird benutzt, um Bilder mit großem Detailreichtum zu erstellen. Grundsätzlich zeigt der C64 direkt einen 8000 Byte großen Speicherbereich auf dem Bildschirm an, wenn er sich im Bit-Map-Modus befindet.

Im Bit-Map-Modus können Sie direkt steuern, ob ein einzelner Punkt auf dem Bildschirm an- oder abgeschaltet ist. Mit dem C64 stehen Ihnen zwei verschiedene Arten von Bit-Mapping zur Verfügung:

1) Standard-Bit-Map-Modus (Hires) (320 x 200 Punkte)
2) Mehrfarben Bit-Map-Modus (160 x 200 Punkte)

Beim Standard-Bit-Mapping ist zwar die Auflösung größer, es stehen jedoch weniger Farbmöglichkeiten zur Verfügung. Beim Mehrfarben-Bit-Mapping wird eine geringe horizontale Auflösung durch die Möglichkeit wettgemacht, mehrere Farben in einem 8 x 8-Punktefeld unterzubringen.

Standard-Bit-Mapping mit hoher Auflösung

Beim Standard-Bit-Mapping haben Sie eine Auflösung von 320 x 200 Punkten und können in jedem 8 x 8-Punktebereich zwischen zwei Farben wählen. Zum Einschalten des Bit-Mapping-Betriebs wird Bit 5 des VIC-II-Kontrollregisters in Adresse 53265 ($D011 in HEX) auf 1 gesetzt. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53265, PEEK(53265) OR 32
```

Zum Abschalten dieser Betriebsart wird dieses Bit gelöscht. Hierzu dient folgende Anweisung:

```
POKE 53265, PEEK (53265) AND 223
```

Bevor wir uns nun im einzelnen mit dem Bit-Map-Modus beschäftigen, müssen wir zuvor ein weiteres Problem lösen: die Plazierung des Bit-Mapping-Bereichs.

Funktionsweise

Wenn Sie noch den Abschnitt über programmierbare Zeichen in Erinnerung haben, werden Sie sich erinnern, daß Sie das Bitmuster eines im RAM gespeicherten Zeichens beliebig wählen können. Genauso wie Sie ein auf dem Bildschirm angezeigtes Zeichen ändern können, können Sie auch einen einzelnen Punkt ändern. Dies ist das Grundmerkmal der Bit-Mapping-Technik.

Der gesamte Bildschirm ist nun mit programmierbaren Zeichen belegt. Ihre Änderungen erfolgen direkt in dem Speicher, vom dem diese programmierbaren Zeichen ihre Muster erhalten, nämlich der Bit-Map.

Jede Adresse im Bildschirmspeicher, die im Normalmodus für die Steuerung der Zeichenwiedergabe benutzt wurde, wird nun für die Farbinformation herangezogen. So wird nun durch das POKEn einer 1 in Speicherplatz 1024 nicht mehr ein »A« links oben auf dem Bildschirm angezeigt, sondern dadurch werden nun die Farben der Bits in der linken oberen Ecke des Bildschirms gesteuert.

Beim Bit-Mapping-Betrieb kommen die Farben der 1000 Bildschirmfelder also nicht vom Farbspeicher wie im Normalmodus, sie werden vielmehr aus dem Bildschirmspeicher genommen. Die oberen 4 Bit des Bildschirmspeichers legen die Farben der Bits fest, die im dazugehörenden Bit-

Map-Bereich auf 1 gesetzt sind. Die unteren 4 Bit enthalten die Farben für jedes Bit, das dort auf 0 gesetzt ist.
Beispiel: Geben Sie folgendes ein:

```
5 BASE=2*4096:POKE 53272, PEEK (53272) OR 8 : REM
  PUT BIT MAP AT 8192
10 POKE 53265, PEEK (53265) OR 32: REM ENTER BIT
  MAP-MODUS EINSCHALTEN
```

Geben Sie nun zum Ausführen des Programms RUN ein.
Auf dem Bildschirm erscheint nichts Brauchbares, stimmt's? Wie der »normale« Bildschirm, muß auch der Hires-Bildschirm zuvor gelöscht werden. In diesem Fall funktioniert das leider nicht durch CLR. Sie müssen vielmehr den Speicherbereich löschen, den Sie für Ihre programmierbaren Zeichen benutzen, also die Bit-Map. Drücken Sie die Tasten <RUN/STOP RESTORE> und fügen dann zum Löschen des Hires-Bildschirms folgende Zeilen in Ihr Programm ein:

```
20 FOR I = BASE TO BASE + 7999 : POKE I, 0:NEXT:REM
  BIT MAP LOESCHEN
30 FOR I = 1024 TO 2023 : POKE I, 3:NEXT:REM
  FARBEN AUF ZYAN UND SCHWARZ SETZEN
```

Geben Sie nun erneut RUN ein. Der Bildschirm wird nun gelöscht und die grünblaue Farbe (Zyan) auf dem ganzen Bildschirm angezeigt. Nun wollen wir einzelne Punkte auf dem Hires-Bildschirm ein- und ausschalten.

Um einen Punkt zu setzen (einzuschalten) oder zu löschen (auszuschalten), müssen Sie wissen, wie Sie das richtige Bit im Zeichenspeicher (der Bit-Map) finden, das auf 1 gesetzt werden soll. Sie müssen dazu das zu ändernde Zeichen, die Zeichenreihe sowie das entsprechende Bit dieser Reihe finden. Für diese Berechnung benötigen Sie eine Gleichung.

Wir benutzen X und Y für die horizontale bzw. vertikale Punktposition. Der Punkt, an dem X gleich 0 und Y gleich 0 ist, befindet sich oben links auf dem Bildschirm. Rechts davon liegende Punkte haben höhere X-Werte und alle Punkte, die darunter liegen, höhere Y-Werte. Der Bildschirm ist also so geordnet, wie es Bild 3 zeigt.



Bild 3. Die Organisation des Bildschirms im hochauflösenden Grafikmodus

Jeder Punkt hat eine X- und eine Y-Koordinate. In diesem Koordinatensystem läßt sich die Lage jedes Punktes auf dem Bildschirm leicht beschreiben.

Bild 4 zeigt Ihnen die Reihenfolge der Bytes auf dem Bildschirm.

Die programmierbaren Zeichen der Bit-Map sind in 25 Reihen mit je 40 Spalten angeordnet. Dies ist zwar für den Textaufbau eine gute Methode, erschwert jedoch die Programmierung im Bit-Map-Modus. Durch nachstehende Gleichung läßt sich ein Punkt in der Bit-Map-Anzeige leichter steuern:

Der Anfang des Bildschirm-Speicherbereichs wird als »Basis« bezeichnet. Die Reihenzahl (von 0 bis 24) Ihres Punktes ist:

```
ROW = INT(Y/8): /REM Es gibt 320 Byte pro Zeile
```

Die Zeichenposition dieser Zeile (von 0 bis 39) lautet:

```
CHAR = INT(X/8) :REM Es gibt 8 Byte pro Zeile
```

Die Zeile dieser Zeichenposition (von 0 bis 7) lautet:

```
LINE = Y AND 7
```

0 (0–7)	1 (8–15)	…	39 (312–319)	Y Zeile	
Byte 0	Byte 8		Byte 312	0	0
Byte 1	Byte 9		Byte 313	1	
Byte 2	Byte 10		Byte 314	2	
Byte 3	Byte 11		Byte 315	3	
Byte 4	Byte 12		Byte 316	4	
Byte 5	Byte 13		Byte 317	5	
Byte 6	Byte 14		Byte 318	6	
Byte 7	Byte 15		Byte 319	7	
Byte 320	Byte 328		Byte 632	8	1
Byte 321	Byte 329		Byte 633	9	
Byte 322	Byte 330		Byte 634	10	
Byte 323	Byte 331		Byte 635	11	
Byte 324	Byte 332		Byte 636	12	
Byte 325	Byte 333		Byte 637	13	
Byte 326	Byte 334		Byte 638	14	
Byte 327	Byte 335		Byte 639	15	
Byte 640				16	
Byte 641				17	
				18	
				190	
				191	
Byte 7680	Byte 7688		Byte 7992	192	24
Byte 7681	Byte 7689		Byte 7993	193	
Byte 7682	Byte 7690		Byte 7994	194	
Byte 7683	Byte 7691		Byte 7995	195	
Byte 7684	Byte 7692		Byte 7996	196	
Byte 7685	Byte 7693		Byte 7997	197	
Byte 7686	Byte 7694		Byte 7998	198	
Byte 7687	Byte 7695		Byte 7999	199	

Bild 4. Der Zusammenhang zwischen Bildschirm und Bit-Map

Das Bit dieses Bytes ist:

```
BIT = 7-(X AND 7)
```

Nun setzen wir diese Gleichungen zusammen. Das Byte, in dem der Punkt (X,Y) liegt, wird wie folgt berechnet:

```
BYTE = BASE + ROW*320 + CHAR*8 + LINE
```

Um ein beliebiges Bit im Gitter mit Koordinaten (X,Y) einzuschalten, verwenden Sie diese Zeile:

```
POKE BYTE, PEEK(BYTE) OR 2↑ BIT
```

Wir fügen diese Berechnungen in ein Programm ein. In Listing 6 zeichnet der C 64 eine Sinuskurve.

Durch die Gleichung in Zeile 60 werden die Werte für die Sinusfunktion von +1 bis –1 in die Y-Werte 10 bis 170 umgeändert. In den Zeilen 70 bis 100 werden das Zeichen, die

```
5 BASE=2*4096:POKE 53272,PEEK(53272) OR 8
  :REM BITMAP AB 8192                           <005>
10 POKE 53265,PEEK(53265) OR 32:REM BITMAP
   MODUS EINSCHALTEN                            <037>
20 FOR I=BASE TO BASE+7999:POKE I,0:NEXT:R
   EM BITMAP LOESCHEN                           <130>
30 FOR I=1024 TO 2023:POKE I,3:NEXT:REM FA
   RBEN AUF CYAN UND SCHWARZ                    <201>
50 FOR X=0 TO 319 STEP .5:REM FUNKTION FUE
   LLT BILDSCHIRM                               <027>
60 Y=INT(90+80*SIN(X/10))                       <165>
70 CH=INT(X/8)                                  <109>
80 RO=INT(Y/8)                                  <107>
85 LN=Y AND 7                                   <212>
90 BY=BASE+RO*320+8*CH+LN                       <195>
100 BI=7-(X AND 7)                              <162>
110 POKE BY,PEEK(BY) OR (2↑BI)                  <008>
120 NEXT X                                      <068>
125 POKE 1024,16                                <109>
130 GOTO 130                                    <106>
```

Listing 6. Zeichnen einer Sinuskurve im hochauflösenden Grafikmodus

```
5 BASE=2*4096:POKE 53272,PEEK(53272) OR 8
  :REM BITMAP AB 8192                          <005>
10 POKE 53265,PEEK(53265) OR 32:REM BITMAP
   MODUS EINSCHALTEN                           <037>
20 FOR I=BASE TO BASE+7999:POKE I,0:NEXT:R
   EM BITMAP LOESCHEN                          <130>
30 FOR I=1024 TO 2023:POKE I,3:NEXT:REM FA
   RBEN AUF CYAN UND SCHWARZ                   <201>
50 FOR X=0 TO 160:REM FUNKTION FUELLT HALB
   EN BILDSCHIRM                               <079>
55 Y1=100+SQR(160*X-X*X)                       <003>
56 Y2=100-SQR(160*X-X*X)                       <013>
60 FOR Y=Y1 TO Y2 STEP Y1-Y2                   <036>
70 CH=INT(X/8)                                 <109>
80 RO=INT(Y/8)                                 <107>
85 LN=Y AND 7                                  <212>
90 BY=BASE+RO*320+8*CH+LN                      <195>
100 BI=7-(X AND 7)                             <162>
110 POKE BY,PEEK(BY) OR (2↑BI)                 <008>
114 NEXT Y                                     <070>
120 NEXT X                                     <068>
125 POKE 1024,16                               <109>
130 GOTO 130                                   <106>
```

Listing 7. Ein Halbkreis wird berechnet und gezeichnet

Reihe, das Byte und zugehörige Bit berechnet. Zeile 125 signalisiert, daß das Programm beendet ist, indem sich in der oberen linken Bildschirmecke die Farbe ändert. Durch Zeile 130 wird das Programm in eine unendliche Schleife geführt. Nach dem Betrachten der Grafik drücken Sie einfach <RUN/STOP RESTORE>. Als weiteres Beispiel ändern wir das Sinuskurvenprogramm so, daß ein Halbkreis angezeigt wird. So ergibt sich Listing 7.

Mehrfarben-Bit-Mapping

Wie beim Mehrfarben-Modus der Zeichen können auch beim Mehrfarben-Bit-Mapping in jedem 8 x 8-Bereich der Bit-Map bis zu vier verschiedene Farben benutzt werden.

Ebenso wie dort wird die horizontale Auflösung von 320 auf 160 Punkte reduziert. Wie bei der Hochauflösungsgrafik dient auch hier der Inhalt einer Bit-Map als Quelle dessen, was der Bildschirm zeigt.

Zum Einschalten dieses Modus wird Bit 5 von Adresse 53265 ($D011) und Bit 4 in Adresse 53270 ($D016) auf 1 gesetzt. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53265, PEEK(53265) OR 32 : POKE 53270,
PEEK(53270) OR 16
```

Zum Ausschalten des Mehrfarben-Bit-Map-Modus wird das Bit wieder gelöscht. Hierzu dient folgende POKE-Anweisung:

```
POKE 53265,PEEK(53265) AND 223: POKE 53270,PEEK
(53270) AND 239
```

Wie beim Standard-Bit-Map-Modus (Hires) besteht eine 1:1-Entsprechung zwischen dem für die Anzeige benutzten 8000-Byte-Speicherbereich und der Bildschirm-Darstellung. Die horizontalen Punkte sind jedoch immer 2 Bit breit. Jeweils 2 Bit im Anzeigenspeicher bilden einen Punkt, der eine von vier Farben haben kann:

BITS	FARBINFORMATION KOMMT VON
00	Hintergrundfarbe #0 (Bildschirmfarbe)
01	Oberen 4 Bit des Bildschirmspeichers
10	Unteren 4 Bit des Bildschirmspeichers
11	Farbnibble (Nibble = ½ Byte = 4 Bit)

Kontinuierliches Verschieben

Der VIC-II-Chip steuert auch das »Scrollen« des Bildschirmes. Darunter versteht man ein Verschieben (Bildschirm-Rollen) des kompletten Bildschirms in eine Richtung (horizontal oder vertikal) um einen frei wählbaren Bildpunkt (Pixel).

Die Bewegung kann entweder nach oben, unten, links oder rechts erfolgen. Hierdurch werden neue Informationen auf einer Seite angezeigt und gleichzeitig verschwinden andere Zeichen auf der gegenüberliegenden Seite.

Auch wenn der VIC-II-Chip Ihnen viele Aufgaben abnimmt, muß dieses Verschieben doch über ein Maschinensprache-Programm erfolgen. Über den VIC-II-Chip kann der Video-Bildschirm in eine beliebige von acht horizontalen und acht vertikalen Positionen gebracht werden. Die Positionierung wird über die VIC-II-Register zum Bildschirmrollen (genannt Scroll-Register) gesteuert. Der VIC-II-Chip hat auch einen 38-Spalten- und 24-Reihen-Modus. Die kleineren Bildschirmgrößen geben Ihnen einen Platz für die neuen Daten, die beim Verschieben gebraucht werden. Gehen Sie für das Verschieben wie folgt vor:

1) Den Bildschirm verkleinern (der Rahmen wird breiter).
2) Das Scroll-Register auf den Maximalwert stellen (oder auf den Minimalwert je nach Richtung des Rollens).
3) Die neuen Daten in den geeigneten Bildschirmbereich eingeben.
4) Das Scroll-Register vergrößern (oder verkleinern), bis es den Maximalwert (oder Minimalwert) erreicht.
5) Zu diesem Zeitpunkt den gesamten Bildschirm um ein Zeichen in Verschieberichtung rollen.
6) Nun zu Schritt 2 zurückgehen.

Diese Schritte werden nun im einzelnen erläutert.

Um in den 38-Spalten-Modus zu gehen, wird Bit 3 von Adresse 53270 ($D016) auf 0 gesetzt. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53270, PEEK (53270) AND 247
```

Zur Rückkehr in den 40-Spalten-Modus ist dieses Bit wieder zu setzen. Hierzu dient folgende POKE-Anweisung:

```
POKE 53270, PEEK (53270) OR 8
```


Für den 24-Reihen-Modus wird Bit 3 von Adresse 53265 ($D011) gelöscht. Das kann geschehen durch:

```
POKE 53265, PEEK (53265) AND 247
```

Zur Rückkehr in den 25-Reihen-Modus muß man Bit 3 wieder setzen:

```
POKE 53265, PEEK (53265) OR 8
```

Beim Verschieben in X-Richtung versetzt man den VIC-II-Chip in den erwähnten 38-Spalten-Modus. Hierdurch wird für die neuen Daten Platz geschaffen. Beim Verschieben nach links werden die neuen Daten rechts eingegeben. Beim Schreiben nach rechts erscheinen die neuen Daten entsprechend auf der linken Seite. Bitte beachten Sie, daß der Bildschirmspeicher noch 40 Spalten hat. Lediglich 38 sind jedoch sichtbar.

Beim Verschieben in Y-Richtung verwendet man den 24-Reihen-Modus. Beim Rollen nach oben gibt man die neuen Daten in die letzte Reihe ein. Beim Rollen nach unten erscheinen die neuen Daten entsprechend in der ersten Reihe. Beim X-Verschieben sind unsichtbare Bereiche auf beiden Bildschirmseiten. Beim Y-Verschieben existiert nur ein unsichtbarer Bereich.

Ist das Y-Scroll-Register auf 0 gesetzt, dann ist die erste Zeile unsichtbar und bereit für neue Daten, enthält es dagegen eine 7, so ist die letzte Reihe unsichtbar. Das Register zum Scrollen in X-Richtung befindet sich in den Bits 0 bis 2 der Adresse 53270 ($D016 HEX).

Auch hier dürfen auf jeden Fall nur diese Bits verändert werden. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53270,(PEEK(53270) AND 248)+X
```

wobei »X« die X-Bildschirmposition 0 bis 7 ist.

Das Y-Scroll-Register wird durch die Bits 0 bis 2 der Adresse 53265 ($D011 HEX) gebildet. Auch hierbei dürfen Sie wieder nur diese Bits verändern. Dazu dient folgende POKE-Anweisung:

```
POKE 53265, (PEEK (53265) AND 248)+Y
```

wobei »Y« die Y-Bildschirmposition 0 bis 7 angibt.

```
10 POKE 53265,PEEK(53265) AND 247        :R
   EM 24-ZEILEN-MODUS                  <141>
20 PRINT CHR$(147)
      :REM BILDSCHIRM LOESCHEN         <048>
30 FOR X=1 TO 24:PRINT CHR$(17);:NEXT
      :REM CURSOR AN UNTEREN RAND      <055>
40 POKE 53265,(PEEK(53265) AND 248) + 7 :R
   EM POSITION FUER 1.SCROLL           <121>
50 PRINT"HALLO";                       <013>
60 FOR P=6 TO 0 STEP -1                <186>
70 POKE 53265,(PEEK(53265) AND 248) + P <036>
80 FOR X=1 TO 50:NEXT                  :R
   EM VERZOEGERUNGSSCHLEIFE            <110>
90 NEXT P:GOTO 40                      <223>
```

Listing 8. Ein Beispiel für vertikales Text-Scrolling

Um den Text von unten auf den Bildschirm zu scrollen, müssen die unteren 3 Bit von Adresse 53265 auf Werte von 0 bis 7 gesetzt, weitere Daten in die abgedeckte Zeile unten auf den Bildschirm eingegeben und danach der Vorgang wiederholt werden.

Ändert man die Verschiebebits mit der Schrittweite von -1, so wird der Text in entgegengesetzter Richtung bewegt.

Das eben Schritt für Schritt besprochene Textrollen vom unteren Bildschirmrand her, zeigt Ihnen Listing 8.

Hilfreiche Geister: Sprites

Ein Sprite ist ein besonderer Typ von frei definierbaren Zeichen, die an beliebiger Stelle auf dem Bildschirm angezeigt werden können. Sprites werden direkt vom VIC-II-Chip verwaltet. Sie brauchen lediglich für jedes Sprite festzulegen, »wie es aussehen«, »welche Farbe es haben«, und »wo es auf dem Bildschirm plaziert werden soll«. Der VIC-II-Chip erledigt für Sie den Rest!

Sprites können zusammen mit jedem beliebigen Grafik-Modus, Bit-Mapping, Zeichen, Mehrfarben-Modus benutzt werden. Eine Sprite-Definition enthält die Farbe, den Modus (Hires oder Mehrfarben) und die Form.

Vom VIC-II-Chip können automatisch gleichzeitig jeweils acht Sprites verwaltet werden. Weitere Sprites sind mittels sogenannter Raster-Interrupt-Techniken in Maschinensprache steuerbar.

Sprites haben folgende Merkmale:
1) Punktgröße 24 x 21 (horizontal x vertikal)
2) Farbsteuerung für jedes Sprite
3) Sprites im Mehrfarbenmodus
4) Vergrößerung (2mal) in horizontaler und/oder vertikaler Richtung
5) Wahlmöglichkeit: Sprites vor oder hinter dem Hintergrund
6) Wahl der Reihenfolge, in der Sprites »hintereinander« angeordnet sind.
7) Sprite-Sprite-Kollisionserkennung
8) Kollisionserkennung zwischen Sprite und Zeichen.

Auf diese Weise lassen sich Spiele einfach programmieren. Da die Sprites durch das Betriebssystem unterstützt werden, kann ein gutes Spiel sogar in Basic geschrieben werden!

Vom VIC-II-Chip werden acht Sprites unterstützt. Sie sind von 0 bis 7 numeriert. Jedes Sprite hat seinen eigenen Speicherbereich für das Bitmuster, seine Positions- und Farbregister sowie seine eigenen Bits zur Erkennung von Kollisionen und zum Ein- und Ausschalten.

Sprite-Definition

Sprites werden genau wie programmierbare Zeichen definiert. Da ein Sprite jedoch größer ist, werden mehr Bytes benötigt. Jedes Sprite besteht aus 24 x 21 oder 504 Punkten. Für die Definition eines Sprite werden also 63 Byte (504/8) benötigt.

Die 63 Byte sind in 21 Reihen zu je 3 Byte angeordnet. Eine Sprite-Definition sieht folgendermaßen aus:

BYTE 0	BYTE 1	BYTE 2
BYTE 3	BYTE 4	BYTE 5
BYTE 6	BYTE 7	BYTE 8
..	..	..
..	..	..
..	..	..
BYTE 60	BYTE 61	BYTE 62

Auch bei der Betrachtung der Sprite-Definition auf Bit-Ebene läßt sich erkennen, wie Sprites erstellt werden. Auf dem Sprite-Entwurfsblatt (Bild 5) lassen sich Sprites sofort in die einzelnen Byte-Werte umrechnen, die man in den DATA-Zeilen in Basic benötigt.

Bei einem Standardsprite (Hires) wird jedes auf 1 gesetzte Bit in der entsprechenden Sprite-Vordergrundfarbe angezeigt. Jedes auf 0 gesetzte Bit ist transparent, so daß der dahinterliegende Wert sichtbar wird. Dies entspricht der Situation beim Standardzeichen.

Mehrfarbige Sprites entsprechen in ihrer Bit-Struktur mehrfarbigen Zeichen. Horizontal wird die Punktauflösung zugunsten der Farbauflösung verschlechtert. Sie beträgt nun 12 x 21 Punkte (horizontal x vertikal). Jeder Punkt im Sprite wird doppelt so breit, dafür sind nun vier Farben darstellbar.

Sprite-Pointer

Jedes Sprite benötigt 64 Byte für seine Definition. Das letzte Byte ist ein Platzhalter. Auf diese Weise können Sie leicht berechnen, wo sich Ihre Sprite-Definition im Speicher befindet, da 64 Byte eine gerade Zahl darstellen und im Binärsystem eine gerade Potenz, nämlich 2^6.

Jedem der acht Sprites dient ein Byte als Sprite-Zeiger (oder auch Sprite-Pointer). Dieser Sprite-Pointer gibt an, wo sich die Sprite-Definition im Speicher befindet. Diese 8 Byte sind stets die letzten 8 Byte vom 1-KByte-Bereich des Bildschirmspeichers. Im Normalfall beginnen sie daher bei Adresse 2040 ($07F8). Beim Verschieben des Bildschirms verändert sich jedoch auch die Lage des Sprite-Pointers.

Jeder Sprite-Pointer kann eine Zahl zwischen 0 und 255 aufnehmen. Diese Zahl zeigt auf das erste Byte der Definition für das Sprite. Da jede Sprite-Definition 64 Byte benötigt, bedeutet dies, daß der Pointer auf jeden Platz im 16-KByte-Speicherbereich zeigen kann, der für den VIC-II-Chip zugänglich ist (da 256*64 = 16 KByte).

Wenn der Sprite-Pointer #0 an Adresse 2040 z.B. die Zahl 14 enthält, bedeutet dies, daß die Definition des Sprite 0 bei Adresse 14 * 64 = 896 beginnt, das heißt im Kassettenpuffer. Dies wird anhand folgender Gleichung deutlich:
SPRITEADRESSE = (BANK*16384) + (WERT DES SPRITE-POINTER* 64)
wobei »BANK« einen der 16-KByte-Speicherbereiche bezeichnet, auf die der VIC-II-Chip zugreifen kann und die von 0 bis 3 durchnumeriert sind. Diese Gleichung gibt den Anfang der 64 Byte des Sprite-Definitionssatzes an. Wenn der VIC-II-Chip auf BANK 0 oder BANK 2 zugreift, ist in einigen Speicherplätzen ein Spiegelbild des Zeichensatzes (wie bereits erwähnt) vorhanden. Hier können keine Sprite-Definitionen stehen.

Einschalten der Sprites

Das VIC-II-Steuerregister in Adresse 53269 ($D015) ist das Sprite-Aktivierungsregister. Jedes Sprite hat in diesem Register ein Bit, das steuert, ob das Sprite EIN oder AUS ist. Das Register sieht folgendermaßen aus:

SPALTEN-NR.	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Summen (für Data-Zeilen)		
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
BITWERTE (EIN=1)	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	1. Byte	2. Byte	3. Byte
Reihe 0																											
Reihe 2																											
Reihe 3																											
Reihe 4																											
Reihe 5																											
Reihe 6																											
Reihe 7																											
Reihe 8																											
Reihe 9																											
Reihe 10																											
Reihe 11																											
Reihe 12																											
Reihe 13																											
Reihe 14																											
Reihe 15																											
Reihe 16																											
Reihe 17																											
Reihe 18																											
Reihe 19																											
Reihe 20																											

Bild 5. Ein Definitionsblatt, das Sie zum Entwurf eigener Sprites kopieren können

$D015	7	6	5	4	3	2	1	0

Um beispielsweise Sprite 1 einzuschalten, muß das Bit 1 gesetzt werden. Dies geschieht durch folgende POKE-Anweisung:

```
POKE 53269, PEEK (53269) OR 2↑1
```

Allgemeiner formuliert gilt:

```
POKE 53269, PEEK (53269) OR (2↑SN)
```

wobei SN die Spritenummer von 0 bis 7 ist.

Ausschalten der Sprites

Ein Sprite wird ausgeschaltet, indem sein Bit im VIC-II-Steuerregister bei 53269 ($D015 HEX) gelöscht wird. Dies geschieht durch die POKE-Anweisung:

```
POKE 53269, PEEK (53269) AND (255 - 2↑SN)
```

wobei SN die Spritenummer von 0 bis 7 angibt.

Farben

Ein Sprite kann eine von 16 möglichen Farben haben. Jedes Sprite hat sein eigenes Sprite-Farbregister. Die Farbregister haben folgende Adressen (siehe Tabelle 8):

ADRESSE		BESCHREIBUNG
53287	($D027)	FARBREGISTER VON SPRITE 0
53288	($D028)	FARBREGISTER VON SPRITE 1
53289	($D029)	FARBREGISTER VON SPRITE 2
53290	($D02A)	FARBREGISTER VON SPRITE 3
53291	($D02B)	FARBREGISTER VON SPRITE 4
53292	($DO2C)	FARBREGISTER VON SPRITE 5
53293	($D02D)	FARBREGISTER VON SPRITE 6
53294	($D02E)	FARBREGISTER VON SPRITE 7

Tabelle 8. Die Farbregister der Sprites

Alle Punkte des Sprite werden in der Farbe angezeigt, deren Wert im Sprite-Farbregister enthalten ist. Nicht gesetzte Punkte des Sprite sind transparent, so daß die hinter diesem Sprite liegenden Werte (normalerweise der Hintergrund) zu sehen sind.

Mehrfarbenmodus

Im Mehrfarbenmodus kann jedes Sprite bis zu vier verschiedene Farben haben. So wie bei den anderen Mehrfarbenmodi ist jedoch auch hier die horizontale Auflösung auf die Hälfte reduziert. Das heißt, beim Arbeiten im Mehrfarbenmodus (wie bei Zeichen im Mehrfarbenmodus) enthält ein Sprite horizontal nicht mehr 24 Punkte, sondern nur noch 12 Punkte. Jedes Punktepaar wird Bitpaar genannt. Stellen Sie sich jedes Bitpaar (Punktepaar) als einen einzelnen Punkt in Ihrem Gesamt-Sprite vor, wenn Sie die Farben für die Punkte in Ihren Sprites wählen.

In der Tabelle 9 finden Sie die Bitpaar-Kombinationen, die Sie zum Einschalten der vier Farben für die Sprites benötigen:

BITPAAR	BESCHREIBUNG
00	TRANSPARENT, BILDSCHIRMFARBE
01	SPRITE-MEHRFARBENREGISTER #0 (53285) ($D025)
10	SPRITE-FARBENREGISTER
11	SPRITE-MEHRFARBENREGISTER #1 (53286) ($d026)

Tabelle 9. Sprite-Bit-Paare und ihre Farbquellen

Wählen des Mehrfarbenmodus für ein Sprite

Den Mehrfarbenmodus für ein Sprite schalten Sie ein, indem Sie das zum Sprite gehörende Bit in der Adresse 53276 ($D01C) auf 1 setzen. Dies geschieht durch folgende POKE-Anweisung:

POKE 53276, PEEK (53276) OR (2↑SN)
wobei SN die Sprite-Nummer angibt (0 bis 7).

Vergrößerte Sprites

Der VIC-II-Chip hat die Fähigkeit, ein Sprite in vertikaler und/oder horizontaler Richtung zu vergrößern. Bei der Ausdehnung wird jeder Punkt im Sprite doppelt so breit oder hoch. Die Auflösung nimmt nicht zu; das Sprite wird lediglich größer.

Um ein Sprite in horizontaler Richtung zu strecken, ist das dazugehörende Bit im VIC-II-Steuerregister in Adresse 53277 ($D01D HEX) auf 1 zu setzen. Durch folgende POKE-Anweisung wird ein Sprite in X-Richtung vergrößert:
POKE 53277, PEEK (53277) OR (2↑SN)
wobei SN die Sprite-Nummer (0 bis 7) angibt.

Durch folgende POKE-Anweisung wird ein Sprite in X-Richtung wieder verkleinert:
POKE 53277, PEEK (53277) AND (255-2↑SN)
wobei SN die Sprite-Nummer von 0 bis 7 angibt. Dadurch wird das zum Sprite gehörende Bit wieder gelöscht.

Die vertikale Streckung eines Sprite geschieht durch Setzen des korrespondierenden Bits in Speicherstelle 53271 ($D017):
POKE 53271, PEEK (53271) OR (2↑SN)
wobei SN die Sprite-Nummer von 0 bis 7 angibt.

Bezugspunkt eines Sprites

Die Verkleinerung bis zur normalen vertikalen Ausdehnung erfolgt durch Löschen des Bits:
POKE 53271, PEEK (53271) AND (255-2↑SN)
wobei SN die Sprite-Nummer von 0 bis 7 angibt.

Sprite-Positionierung

Nachdem Sie ein Sprite konstruiert haben, können Sie es auf dem Bildschirm bewegen. Hierzu benutzt der C64 drei Positionsregister:

1) Sprite X-Positionsregister
2) Sprite Y-Positionsregister
3) Höchstes Bit des X-Positionsregisters (engl. MSB = Most significant Bit).

Jedes Sprite hat ein X-Positionsregister, ein Y-Positionsregister und ein Bit im MSB-X-Register. Es stehen Ihnen 512 mögliche X- und 256 mögliche Y-Positionen zur Verfügung. Die X- und Y-Positionsregister »arbeiten« paarweise zusammen. Die Adressen von X- und Y-Register erscheinen wie folgt im Speicher: Zunächst das X-Register von Sprite 0, dann das Y-Register für das gleiche Sprite.

Danach folgt das X-Register und dann das Y-Register für Sprite 1 etc.

Nach allen 16 X- und Y-Registern kommt das MSB-X-Register, das für jedes Sprite ein Bit zur Verfügung stellt.

Die Tabelle 10 gibt die Adressen der einzelnen Sprite-Positionsregister an. Sie können auf diese Adressen durch POKE-Anweisungen zugreifen:

Als Bezugspunkt eines Sprite zur Positionsangabe dient die obere linke Ecke des 24 x 21-Punktebereichs, der für ein Sprite zur Verfügung steht. Es spielt herbei keine Rolle, wie viele bzw. wenige Punkte Sie für ein Sprite benutzt haben. Auch wenn nur ein Punkt für das Sprite benutzt wurde, und dieses in der Mitte des Bildschirms stehen soll, müssen Sie für die Positionierung die obere linke Ecke des Sprite als Bezugspunkt verwenden.

Vertikale Positionierung

Die Positionierung in horizontaler Richtung ist etwas schwieriger als die vertikale Positionierung. Daher werden wir uns zunächst mit ihr beschäftigen.

Es gibt 200 sichtbare Punktpositionen, die auf dem Bildschirm in Y-Richtung programmiert werden können. Das Y-Positionsregister der Sprites kann jedoch Zahlen bis zu 255 fassen, so daß Sprites auch aus dem Bildschirm hinaus- bzw. hineinbewegt werden können.

Der erste Wert, bei dem ein Sprite von oben auf dem Bildschirm auftaucht und der für ein in Y-Richtung unvergrößertes Sprite gilt, ist 30. Für ein in Y-Richtung gestrecktes Sprite lautet dieser Wert 9 (da jeder Punkt zweimal so hoch ist und die Ausgangsposition auch hier von der obersten linken Ecke des Sprite berechnet wird, ist hier der entsprechende Wert kleiner).

Der erste Y-Wert, bei dem ein Sprite (vergrößert oder nicht) ganz auf dem Bildschirm erscheint (alle 21 möglichen Zeilen werden angezeigt) lautet 50.

Der letzte Y-Wert, bei dem ein unvergrößertes Sprite noch ganz auf dem Bildschirm vorhanden ist, ist 229. Für ein vergrößertes Sprite lautet der Wert 208.

Der erste Y-Wert, bei dem ein Sprite vollständig vom Bildschirm verschwunden ist, ist 250. In Listing 9 finden Sie ein Beispiel zur Positionierung des Sprite 0.

```
10 PRINT CHR$(147)        :REM BILDSCHIRM L
   OESCHEN                                <038>
20 POKE 2040,13          :REM DATEN SPRITE
   0 AUS BLOCK 13                         <029>
30 FOR I=0 TO 62:POKE 832+I,129:NEXT:REM S
   PRITEDATEN EINTRAGEN                   <044>
40 V=53248               :REM STARTADRESSE
   DES VIDEOCHIP                          <123>
50 POKE V+21,1           :REM SPRITE 0 EINS
   CHALTEN                                <090>
60 POKE V+39,1           :REM SPRITE 0 FARB
   E GEBEN                                <129>
70 POKE V+1,100          :REM SPRITE 0 Y-PO
   SITION                                 <104>
80 POKE V,100:POKE V+16,0:REM SPRITE 0 X-P
   OSITION                                <188>
```

Listing 9. Positionierung des Sprite Nummer 0

Horizontale Positionierung

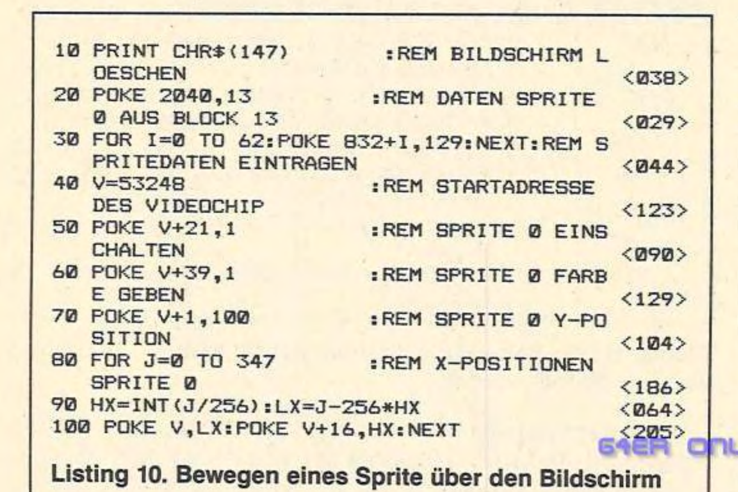
Positionierung in horizontaler Richtung ist komplizierter, da hier mehr als 256 Positionen zur Verfügung stehen. Man

ADRESSE		BESCHREIBUNG	ADRESSE		BESCHREIBUNG
DEZIMAL	HEX.		DEZIMAL	HEX.	
53248	($D000)	X-POSITIONSREGISTER VON SPRITE 0	53257	($D009)	Y-POSITIONSREGISTER VON SPRITE 4
53249	($D001)	Y-POSITIONSREGISTER VON SPRITE 0	53258	($D00A)	X-POSITIONSREGISTER VON SPRITE 5
53250	($D002)	X-POSITIONSREGISTER VON SPRITE 1	53259	($D00B)	Y-POSITIONSREGISTER VON SPRITE 5
53251	($D003)	Y-POSITIONSREGISTER VON SPRITE 1	53260	($D00C)	X-POSITIONSREGISTER VON SPRITE 6
53252	($D004)	X-POSITIONSREGISTER VON SPRITE 2	53261	($D00D)	Y-POSITIONSREGISTER VON SPRITE 6
53253	($D005)	Y-POSITIONSREGISTER VON SPRITE 2	53262	($D00E)	X-POSITIONSREGISTER VON SPRITE 7
53254	($D006)	X-POSITIONSREGISTER VON SPRITE 3	53263	($D00F)	Y-POSITIONSREGISTER VON SPRITE 7
53255	($D007)	Y-POSITIONSREGISTER VON SPRITE 3	53264	($D010)	X-MSB REGISTER
53256	($D008)	X-POSITIONSREGISTER VON SPRITE 4			

Tabelle 10. Die Sprite-Positionsregister

benötigt daher ein neuntes Bit zur Steuerung der X-Position. Durch Hinzunahme des Extrabits hat ein Sprite nun 512 mögliche Positionen in der X-Richtung (links/rechts). Dadurch stehen mehr Positionen zur Verfügung, als auf dem Bildschirm angezeigt werden können, und jedes Sprite kann eine Position von 0 bis 511 haben. Es sind jedoch lediglich die Werte zwischen 24 und 343 auf dem Bildschirm sichtbar. Wenn die X-Position eines Sprite größer als 255 (auf der rechten Bildschirmseite) ist, muß das der Sprite-Nummer entsprechende Bit im MSB-Register (53264) auf 1 gesetzt sein. Wenn die X-Position eines Sprite kleiner als 256 (z.B. auf der linken Bildschirmseite) ist, dann muß das X-MSB-Register dieses Sprite auf 0 gesetzt sein.

Die Bits 0 bis 7 des X-MSB-Registers entsprechen den Sprites 0 bis 7. Durch das Programm (Listing 10) wird ein Sprite über den Bildschirm bewegt.



```
10 PRINT CHR$(147)          :REM BILDSCHIRM L
   OESCHEN                                <038>
20 POKE 2040,13            :REM DATEN SPRITE
   0 AUS BLOCK 13                         <029>
30 FOR I=0 TO 62:POKE 832+I,129:NEXT:REM S
   PRITEDATEN EINTRAGEN                   <044>
40 V=53248                 :REM STARTADRESSE
   DES VIDEOCHIP                          <123>
50 POKE V+21,1             :REM SPRITE 0 EINS
   CHALTEN                                <090>
60 POKE V+39,1             :REM SPRITE 0 FARB
   E GEBEN                                <129>
70 POKE V+1,100            :REM SPRITE 0 Y-PO
   SITION                                 <104>
80 FOR J=0 TO 347          :REM X-POSITIONEN
   SPRITE 0                               <186>
90 HX=INT(J/256):LX=J-256*HX              <064>
100 POKE V,LX:POKE V+16,HX:NEXT           <205>
```


Listing 10. Bewegen eines Sprite über den Bildschirm

Beim Bewegen von vergrößerten Sprites auf die linke Bildschirmseite in X-Richtung soll zu Beginn der Bewegung das Sprite auf der rechten Seite nicht sichtbar sein. Ein erweitertes Sprite ist nämlich größer als der verfügbare Platz auf der linken Bildschirmseite. Das zeigt Ihnen Listing 11.

Die Bilder 6 und 7 erklären die Sprite-Positionierung. Damit können Sie jedes Sprite beliebig positionieren. Durch Bewegung eines Sprite um jeweils eine einzelne Punktposition wird eine fließende Bewegung möglich.

```
10 PRINT CHR$(147)          :REM BILDSCHIRM L
   OESCHEN                                <038>
20 POKE 2040,13            :REM DATEN SPRITE
   0 AUS BLOCK 13                         <029>
30 FOR I=0 TO 62:POKE 832+I,255:NEXT:REM S
   PRITEDATEN EINTRAGEN                   <026>
40 V=53248                 :REM STARTADRESSE
   DES VIDEOCHIP                          <123>
50 POKE V+21,1             :REM SPRITE 0 EINS
   CHALTEN                                <090>
60 POKE V+39,1:POKE V+23,1:POKE V+29,1:REM
    SPRITE 0 FARBE GEBEN UND VERGROESSERN <062>
70 POKE V+1,100            :REM SPRITE 0 Y-PO
   SITION                                 <104>
80 J=488                                  <102>
90 HX=INT(J/256):LX=J-256*HX: IF HX<0 THEN
   80                                     <150>
100 POKE V,LX:POKE V+16,HX                <142>
110 J=J-1:IF J>511 THEN J=0               <149>
120 IF J<488 OR J>348 THEN 90             <133>
```

Listing 11. Das Sprite wird nun vergrößert und ebenfalls über den Bildschirm bewegt

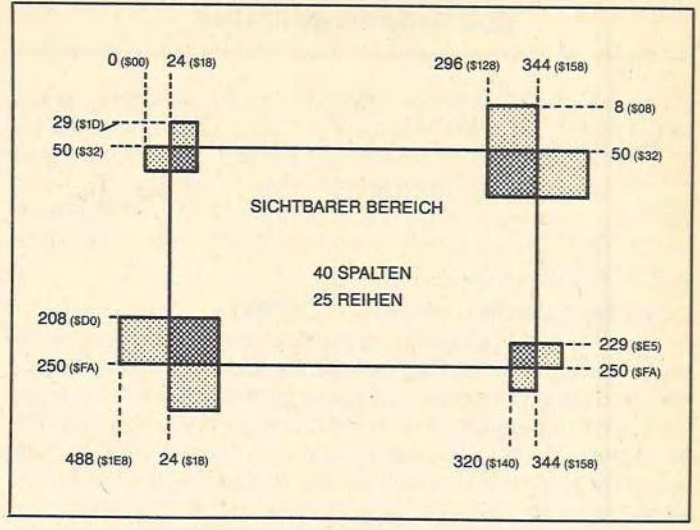


Bild 6. Sprite-Positionen bei normalem Bildschirm

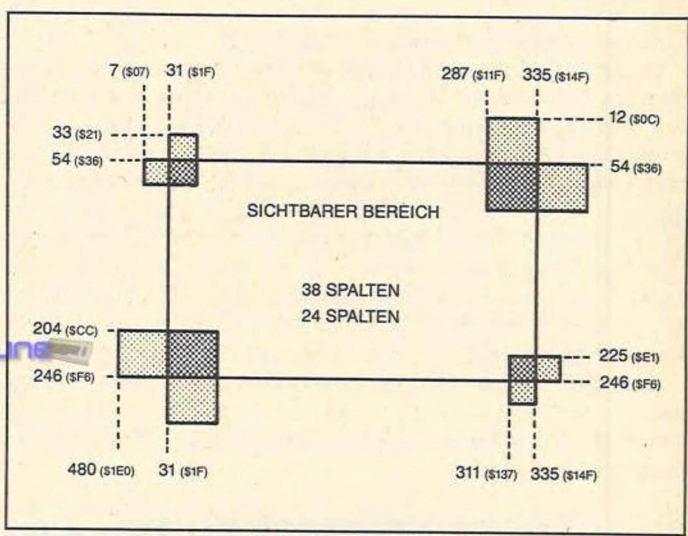


Bild 7. Sprite-Positionen bei verkleinertem Bildschirm

Sprite-Anzeigeprioritäten

Die Wege der verschiedenen Sprites können sich kreuzen. Darüber hinaus können sich Sprites vor oder hinter anderen Objekten auf dem Bildschirm bewegen. Durch diese räumliche Darstellung können Sie bei Spielen einen dreidimensionalen Effekt erzeugen. Die Priorität der einzelnen Sprites ist festgelegt. Sprite 0 hat dabei die oberste, Sprite 1 die nächste usw., so daß Sprite 7 entsprechend die niedrigste Priorität hat. Wenn also Sprite 1 und Sprite 6 einander kreuzen, so erscheint Sprite 1 vor Sprite 6.

Soll ein Sprite im Bildvordergrund erscheinen, so muß es eine niedrigere Nummer erhalten, als das, das im Hintergrund erscheinen soll.

Dabei ist ein »Fenstereffekt« möglich. Hat ein Sprite mit höherer Priorität »Löcher« (Bereiche, in denen die Punkte nicht gesetzt sind), so scheinen Sprites mit niedriger Priorität durch. Das gleiche gilt für Sprites und Hintergrunddaten.

Das Prioritätsverhältnis zwischen Sprites und Bildschirmhintergrund wird durch ein Register in Adresse 53275 ($D01B) gesteuert. In diesem Register ist jedem Sprite ein Bit zugeordnet. Ist das Bit 0, so hat das entsprechende Sprite eine höhere Priorität als der Bildschirmhintergrund, das heißt, das Sprite erscheint vor den Hintergrunddaten. Ist das Bit 1, so hat der Hintergrund Priorität gegenüber dem Sprite. Dieses erscheint dann hinter den Hintergrunddaten.

Kollisionserkennung

Einer der interessanteren Aspekte des VIC-II-Chips ist die Möglichkeit der Kollisionserkennung. Zusammenstöße können zwischen verschiedenen Sprites oder Sprites und einem bestimmten Hintergrund erkannt werden. Zu einer Kollision kommt es, wenn ein »Nicht-0«-Teil eines Sprite einen »Nicht-0«-Teil eines weiteren Sprite oder ein Bildschirmzeichen überlappt.

Kollision zwischen einzelnen Sprites

Eine Kollision zwischen einzelnen Sprites wird vom Computer erkannt und im Register 53278 ($D01E) gekennzeichnet. In diesem Register hat jedes Sprite ein Bit. Ist dieses Bit 1, dann ist das Sprite an einer Kollision beteiligt. Die Bits in diesem Register bleiben bis zum Lesen (PEEK-Anweisung) gesetzt. Nach dem Lesen wird das Register automatisch gelöscht. Der Wert sollte daher sofort in einer Variablen gespeichert werden, bis er verarbeitet wird.

Kollisionen können auch dann auftreten, wenn Sprites ausgeschaltet sind.

Kollision zwischen Sprites und Daten

Zusammenstöße von Sprites mit Bildschirmzeichen werden im Register in Adresse 53279 ($D01F) festgestellt. Auch hier hat jedes Sprite ein Bit. Ist dieses Bit eine 1, dann ist dieses Sprite an einer Kollision beteiligt. Die Bits in diesem Register bleiben bis zum Lesen (PEEK-Anweisung) gesetzt. Auch hier wird das Register nach einem Lesevorgang gelöscht, Sie sollten diesen Wert ebenfalls in einer Variablen zwischenspeichern. Der Multi-Color-Wert 01 wird bei Kollisionen als transparent angesehen, auch wenn er auf dem Bildschirm sichtbar ist. Beim Erstellen eines Hintergrunds sollte daher all das, was nicht zu einer Kollision führen darf, im Mehrfarbenmodus auf 01 gesetzt werden.

Einige Beispiele zur Spriteprogrammierung zeigen die Listings 12, 13 und 14.

Weitere Grafikmöglichkeiten

Über Bit 4 des VIC-II-Steuerregisters an Adresse 53265 ($D011) wird das Ein- bzw. Ausschalten des Bildschirms gesteuert. Ist dieses Bit eingeschaltet (das heißt auf 1 gesetzt), dann ist der Bildschirm normal. Ist Bit 4 ausgeschaltet (=0), dann nimmt der gesamte Bildschirm die Rahmenfarbe an.

Durch folgende POKE-Anweisung wird der Bildschirm weggeblendet. Die Daten gehen nicht verloren, sie werden lediglich nicht mehr angezeigt.

```
POKE 53265, PEEK(53265) AND 239
```

Den normalen Bildschirm erhält man durch:

```
POKE 53265, PEEK(53265)OR 16
```

Durch Ausschalten des Bildschirms wird der Prozessor entlastet und die Programmausführung erfolgt daher etwas schneller.

Rasterregister

Das Rasterregister befindet sich im VIC-II-Chip an Adresse 53266 ($D012). Es erfüllt einen doppelten Zweck. Beim Lesen des Registers werden die unteren 8 Bits der derzeitigen Rasterposition wiedergegeben. Das Bit 9 der Rasterposition befindet sich als Bit 7 in Adresse 53265 ($D011). Dieses Register ist von Bedeutung, wenn man beim Verändern von Bildschirminhalten ein Flackern befürchten muß. Man sollte solch eine Änderung dann so vornehmen, daß die Rasterstrahlposition nicht im sichtbaren Anzeigenbereich liegt, das heißt, wenn die Punktpositionen zwischen 51 und 251 liegen.

Schreibt man in das Rasterregister (einschließlich MSB), wird die zugeordnete Zahl für den Rastervergleich gespeichert. Ist der tatsächliche Rasterwert gleich der Zahl des Rasterregisters, so wird ein Bit im VIC-II-Chip-Interrupt-Register 53273 ($D019) auf 1 gesetzt (EIN).

Steht das richtige Interrupt-Bit wirksam auf 1, so kommt es zu einem Raster-Interrupt (IRQ).

Interrupt-Statusregister

Das Interrupt-Statusregister zeigt den derzeitigen Status einer beliebigen Interrupt-Quelle. So ist der Status von Bit 2 des Interrupt-Registers 1, wenn es zu einer Kollision zwischen zwei Sprites kommt. Das gleiche gilt entsprechend für die in Tabelle 11 aufgeführten Bits 0 und 3. Auch Bit 7 wird bei einem Interrupt auf 1 gesetzt.

Das Interrupt-Statusregister befindet sich in Adresse 53273 ($D019).

SCHALTER	BIT #	BESCHREIBUNG
IRST	0	Gesetzt, wenn derzeitiger Rasterwert gleich gespeichertem Rasterwert.
IMDC	1	Gesetzt durch eine Kollision zwischen Sprite und einem Zeichen auf dem Bildschirm, Zurückstellung durch RESET.
IMMC	2	Gesetzt durch eine Kollision zwischen zwei Sprites, Zurückstellung durch RESET.
ILP	3	Gesetzt bei negativer Flanke am Lightpen-Eingang.
IRQ	7	Wird gesetzt, wenn eines der Bits #0 bis 3 gesetzt ist.

Tabelle 11. Die Bedeutung der einzelnen Bits im Interrupt-Statusregister

Nach dem Setzen eines Interrupt-Bits ist dieses »eingerastet« und muß durch Einschreiben einer 1 in dieses Bit im Interrupt-Register gelöscht werden (= RESET). Hierdurch kann der Interrupt selektiv ohne Störung der anderen Interrupt-Bits gehandhabt werden.

Das Interrupt-Aktivierungsregister befindet sich in Adresse 53274 ($D01A). Dieses Register hat das gleiche Format wie das Interrupt-Statusregister. Wenn das entsprechende Bit im Interrupt-Aktivierungsregister nicht auf 1 gesetzt ist, wird von dieser Quelle kein Interrupt angefordert. Das Interrupt-Statusregister kann noch immer abgerufen werden, es werden jedoch keine Interrupts erzeugt. Um eine Interrupt-Anforderung wirksam zu machen, muß das entsprechende Interrupt-Aktivierungsbit (wie in Tabelle 11 gezeigt) auf 1 gesetzt sein.

Über diese Interrupt-Struktur können Betriebsarten mit geteiltem Bildschirm benutzt werden. So kann z. B. für die eine Hälfte des Bildschirms Bit-Mapping, für die andere Hälfte Text, mehr als 8 Sprite gleichzeitig usw. benutzt werden. Die Interrupts müssen nur richtig gehandhabt werden. Soll die obere Bildschirmhälfte z. B. im Bit-Mapping und die untere mit Text dargestellt werden, muß lediglich das Raster-Vergleichsregister (wie bereits erklärt) für die untere Bildschirmhälfte gesetzt sein. Bei einem Interrupt muß der VIC-II-Chip die Zeichen aus dem ROM nehmen; dann wird das Raster-Vergleichsregister für einen Interrupt am oberen Bildschirmrand eingestellt. Wenn es dort zu einem Interrupt kommt, muß der VIC-II-Chip die Zeichen aus dem RAM (Bit-Mapping) nehmen.

Auf die gleiche Weise können auch mehr als 8 Sprite angezeigt werden. Hierzu ist Basic jedoch leider nicht schnell genug. Beim Arbeiten mit Interrupts müssen Sie also Maschinensprache wählen.

Vorschläge für Bildschirm-Zeichenfarbe-Kombinationen

Bei Farbfernsehgeräten gibt es Einschränkungen hinsichtlich der Fähigkeit, bestimmte Farben nebeneinander

```
10 REM * SPRITE BEISPIEL 1 *                       <201>
20 REM DER HEISSLUFTBALLON                         <031>
30 VIC=13*4096:REM HIER BEGINNEN DIE VIC-R
   EGISTER                                         <157>
35 POKE VIC+21,1:REM EINSCHALTEN SPRITE 0          <114>
36 POKE VIC+33,14:REM HINTERGRUNDFARBE HEL
   LBLAU                                           <120>
37 POKE VIC+23,1:REM SPRITE 0 IN Y VERDOPP
   ELN                                             <096>
38 POKE VIC+29,1:REM UND IN X                      <184>
40 POKE 2040,192:REM SPRITEZEIGER 0 AUF BL
   OCK 192                                         <181>
180 POKE VIC,100:REM X-POSITION SPRITE 0           <215>
190 POKE VIC+1,100:REM UND Y-POSITION              <017>
220 POKE VIC+39,1:REM FARBE SPRITE 0               <090>
250 FOR Y=0 TO 63                                  <114>
300 READ A:REM SPRITEDATEN LESEN                   <026>
310 POKE 192*64+Y,A:REM UND EINTRAGEN              <086>
320 NEXT Y                                         <022>
330 DX=1:DY=1                                      <162>
340 X=PEEK(VIC):REM X-POSITION PRUEFEN             <020>
350 Y=PEEK(VIC+1):REM Y-POSITION TESTEN            <192>
360 IF Y=50 OR Y=200 THEN DY=-DY:REM BEI R
    ANDPOSITION DY UMKEHREN                        <192>
380 IF X=24 AND (PEEK(VIC+16)AND 1)=0 THEN
     DX=-DX:REM WENN SPRITE                        <054>
390 REM AM LINKEN RAND (X<24 UND MSB=0) DA
    NN DX UMKEHREN                                 <219>
400 IF X=40 AND (PEEK(VIC+16)AND 1)=1 THEN
     DX=-DX:REM WENN SPRITE                        <201>
410 REM AM RECHTEN RAND (X=40 UND MSB=1) D
    ANN DX UMDREHEN                                <091>
420 IF X=255 AND DX=1 THEN X=-1:SIDE=1             <023>
430 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                            <213>
440 IF X=0 AND DX=-1 THEN X=256:SIDE=0             <063>
450 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                            <233>
460 X=X+DX:REM ADDIEREN VON DX ZU X                <170>
470 X=X AND 255:REM SICHERSTELLEN DASS X I
    M ERLAUBTEN BEREICH                            <051>
480 Y=Y+DY                                         <131>
485 POKE VIC+16,SIDE                               <070>
490 POKE VIC,X:REM NEUE X-POSITION SPRITE
    0                                              <000>
510 POKE VIC+1,Y:REM NEUER Y-WERT                  <056>
530 GOTO 340                                       <030>
600 REM ***** SPRITE DATEN *****                   <131>
610 DATA 0,127,0,1,255,192,3,255,224,3,231
    ,224                                           <199>
620 DATA 7,217,240,7,223,240,7,217,240,3,2
    31,224                                         <191>
630 DATA 3,255,224,3,255,224,2,255,160,1,1
    27,64                                          <198>
640 DATA 1,62,64,0,156,128,0,156,128,0,73,
    0,0,73,0                                       <076>
650 DATA 0,62,0,0,62,0,0,62,0,0,28,0,0             <111>
```

Listing 12. Sprite-Demo: Ein Ballon fliegt über den Bildschirm

```
10 REM * SPRITE BEISPIEL 2 *                       <205>
20 REM NOCH EINMAL DER HEISSLUFTBALLON             <004>
30 VIC=13*4096:REM HIER BEGINNEN DIE VIC-R
   EGISTER                                         <157>
35 POKE VIC+21,63:REM EINSCHALTEN SPRITE 0
   BIS 5                                           <036>
36 POKE VIC+33,14:REM HINTERGRUNDFARBE HEL
   LBLAU                                           <120>
37 POKE VIC+23,3:REM SPRITE 0 UND 1 IN Y V
   ERDOPPELN                                       <166>
38 POKE VIC+29,3:REM UND IN X                      <192>
40 POKE 2040,192:REM SPRITEZEIGER 0 AUF BL
   OCK 192                                         <181>
50 POKE 2041,193:REM SPRITEYEIGER 1                <095>
60 POKE 2042,192:REM SPRITEZEIGER 2                <175>
70 POKE 2043,193:REM SPRITEZEIGER 3                <255>
80 POKE 2044,192:REM SPRITEZEIGER 4                <072>
90 POKE 2045,193:REM SPRITEZEIGER 5                <152>
100 POKE VIC+4,30:REM X-POSITION SPRITE 2          <151>
110 POKE VIC+5,58:REM Y-POSITION SPRITE 2          <101>
120 POKE VIC+6,65:REM X-POSITION SPRITE 3          <230>
130 POKE VIC+7,58:REM Y-POSITION SPRITE 3          <154>
140 POKE VIC+8,100:REM X-POSITION SPRITE 4         <218>
150 POKE VIC+9,58:REM Y-POSITION SPRITE 4          <207>
160 POKE VIC+10,100:REM X-POSITION SPRITE
    5                                              <159>
170 POKE VIC+11,58:REM Y-POSITION SPRITE 5         <067>
175 PRINT"{WHITE,CLR}"TAB(15)"DIES SIND 2
    HIRES-SPRITES";                                <047>
176 PRINT TAB(55)"EINER UEBER DEM ANDEREN"         <105>
180 POKE VIC,100:REM X-POSITION SPRITE 0           <215>
190 POKE VIC+1,100:REM UND Y-POSITION              <017>
200 POKE VIC+2,100:REM X-POSITION SPRITE 1         <082>
210 POKE VIC+3,100:REM Y-POSITION SPRITE 1         <028>
220 POKE VIC+39,1:REM FARBE SPRITE 0               <090>
230 POKE VIC+41,1:REM FARBE SPRITE 2               <223>
240 POKE VIC+43,1:REM FARBE SPRITE 4               <237>
250 POKE VIC+40,6:REM FARBE SPRITE 1               <005>
260 POKE VIC+42,6:REM FARBE SPRITE 3               <021>
270 POKE VIC+44,6:REM FARBE SPRITE 5               <035>
280 FOR X=192 TO 193:REM SPRITEDEFINITIONE
    N                                              <221>
290 FOR Y=0 TO 63:REM BYTEZAEHLER                  <233>
300 READ A:REM SPRITEDATEN LESEN                   <026>
310 POKE X*64+Y,A:REM UND EINTRAGEN                <232>
320 NEXT Y,X                                       <227>
330 DX=1:DY=1                                      <162>
340 X=PEEK(VIC):REM X-POSITION SPRITE 0 PR
    UEFEN                                          <207>
350 Y=PEEK(VIC+1):REM Y-POSITION TESTEN            <192>
360 IF Y=50 OR Y=200 THEN DY=-DY:REM BEI R
    ANDPOSITION DY UMKEHREN                        <192>
380 IF X=24 AND (PEEK(VIC+16)AND 1)=0 THEN
     DX=-DX:REM WENN SPRITE                        <054>
390 REM AM LINKEN RAND (X<24 UND MSB=0) DA
    NN DX UMKEHREN                                 <219>
400 IF X=40 AND (PEEK(VIC+16)AND 1)=1 THEN
     DX=-DX:REM WENN SPRITE                        <201>
410 REM AM RECHTEN RAND (X=40 UND MSB=1) D
    ANN DX UMDREHEN                                <091>
420 IF X=255 AND DX=1 THEN X=-1:SIDE=3             <025>
430 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                            <213>
440 IF X=0 AND DX=-1 THEN X=256:SIDE=0             <063>
450 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                            <233>
460 X=X+DX:REM ADDIEREN VON DX ZU X                <170>
470 X=X AND 255:REM SICHERSTELLEN DASS X I
    M ERLAUBTEN BEREICH                            <051>
480 Y=Y+DY                                         <131>
485 POKE VIC+16,SIDE                               <070>
490 POKE VIC,X:REM NEUE X-POSITION SPRITE
    0                                              <000>
500 POKE VIC+2,X:REM NEUER X-WERT SPRITE 1         <137>
510 POKE VIC+1,Y:REM NEUER Y-WERT                  <056>
520 POKE VIC+3,Y:REM NEUER Y-WERT SPRITE 1         <035>
530 GOTO 340                                       <030>
600 REM ***** SPRITE DATEN *****                   <131>
610 DATA 0,255,0,3,153,192,7,24,224,7,56,2
    24,14,126,112,14,126,112,14,126,112            <192>
620 DATA 6,126,96,7,56,224,7,56,224,1,56,1
    28,0,153,0,90,0,0,56,0                         <148>
630 DATA 0,56,0,0,0,0,0,0,0,126,0,0,42,0
    ,0,84,0,0,40,0,0                               <001>
640 DATA 0,0,0,0,102,0,0,231,0,0,195,0,1,1
    29,128,1,129,128,1,129,128                     <225>
650 DATA 1,129,128,0,195,0,0,195,0,4,195,3
    2,2,102,64,2,36,64,1,0,128                     <248>
660 DATA 1,0,128,0,153,0,153,0,0,0,0,0,8
    4,0,0,42,0,0,20,0,0                            <110>
```

Listing 13. Mit einem Trick lassen sich auch farbige Hires-Sprites darstellen

anzuzeigen. Bestimmte Kombinationen von Bildschirm und Zeichenfarben führen zu unscharfen Bildern. Tabelle 12 zeigt Ihnen, welche Farbkombinationen Sie besser vermeiden sollten und welche Farben gut miteinander kombiniert werden können.

BILDSCHIRMFARBE \ ZEICHENFARBE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	x	○	x	○	○	●	x	○	○	x	○	○	○	○	○	○
1	○	x	○	x	○	○	○	x	●	○	●	○	○	x	○	○
2	x	○	x	x	●	x	x	○	○	x	○	x	x	x	x	●
3	○	x	x	x	x	●	○	x	x	x	x	●	x	x	●	x
4	○	●	x	x	x	x	x	x	x	x	x	x	x	x	x	●
5	○	●	x	●	x	x	x	x	x	x	x	●	x	○	x	●
6	●	●	x	○	x	x	x	x	x	x	x	x	x	●	○	○
7	○	x	○	x	x	x	●	x	●	○	●	○	○	x	x	x
8	●	○	○	x	x	x	x	○	x	○	x	x	x	x	x	●
9	x	○	x	x	x	x	x	○	○	x	○	x	x	x	x	○
10	●	●	○	x	x	x	x	●	x	○	x	x	x	x	x	●
11	○	○	x	●	x	x	x	○	x	x	x	x	○	○	●	○
12	○	○	●	x	x	x	●	x	x	●	x	○	x	x	x	●
13	○	x	x	x	x	○	●	x	x	x	x	○	x	x	x	x
14	○	○	x	○	x	x	○	x	x	x	x	●	x	x	x	●
15	○	○	○	x	●	●	○	x	x	●	●	○	○	x	●	x

○ = GUT ● = ANNEHMBAR x = SCHLECHT

Tabelle 12. Die Wirkung verschiedener Zeichen-/ Bildschirm-Farbkombinationen


```
10 REM * SPRITE BEISPIEL 3 *                  <209>
20 REM DAS HEISSLUFTMONSTER                   <148>
30 VIC=13*4096:REM HIER BEGINNEN DIE VIC-R
   EGISTER                                    <157>
35 POKE VIC+21,1:REM EINSCHALTEN SPRITE 0     <114>
36 POKE VIC+33,14:REM HINTERGRUNDFARBE HEL
   LBLAU                                      <120>
37 POKE VIC+23,1:REM SPRITE 0 IN Y VERDOPP
   ELN                                        <096>
38 POKE VIC+29,1:REM UND IN X                 <184>
40 POKE 2040,192:REM SPRITEZEIGER 0 AUF BL
   OCK 192                                    <181>
50 POKE VIC+28,1:REM MEHRFARBENMODUS SPRIT
   E 0                                        <145>
60 POKE VIC+37,7:REM FARBREGISTER 0 SETZEN    <027>
70 POKE VIC+38,4:REM FARBREGISTER 1 SETZEN    <028>
180 POKE VIC,100:REM X-POSITION SPRITE 0      <215>
190 POKE VIC+1,100:REM UND Y-POSITION         <017>
220 POKE VIC+39,2:REM FARBE SPRITE 0          <094>
250 FOR Y=0 TO 63                             <114>
300 READ A:REM SPRITEDATEN LESEN              <026>
310 POKE 192*64+Y,A:REM UND EINTRAGEN         <086>
320 NEXT Y                                    <022>
330 DX=1:DY=1                                 <162>
340 X=PEEK(VIC):REM X-POSITION PRUEFEN        <020>
350 Y=PEEK(VIC+1):REM Y-POSITION TESTEN       <192>
360 IF Y=50 OR Y=208 THEN DY=-DY:REM BEI R
    ANDPOSITION DY UMKEHREN                   <064>
380 IF X=24 AND (PEEK(VIC+16)AND 1)=0 THEN
     DX=-DX:REM WENN SPRITE                   <054>
390 REM AM LINKEN RAND (X<24 UND MSB=0) DA
    NN DX UMKEHREN                            <219>
400 IF X=40 AND (PEEK(VIC+16)AND 1)=1 THEN
     DX=-DX:REM WENN SPRITE                   <201>
410 REM AM RECHTEN RAND (X=40 UND MSB=1) D
    ANN DX UMDREHEN                           <091>
420 IF X=255 AND DX=1 THEN X=-1:SIDE=1        <023>
430 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                       <213>
440 IF X=0 AND DX=-1 THEN X=256:SIDE=0        <063>
450 REM UMSCHALTEN AUF ANDERE BILDSCHIRMSE
    ITE                                       <233>
460 X=X+DX:REM ADDIEREN VON DX ZU X           <170>
470 X=X AND 255:REM SICHERSTELLEN DASS X I
    M ERLAUBTEN BEREICH                       <051>
480 Y=Y+DY                                    <131>
485 POKE VIC+16,SIDE                          <070>
490 POKE VIC,X:REM NEUE X-POSITION SPRITE
    0                                         <000>
510 POKE VIC+1,Y:REM NEUER Y-WERT             <056>
520 GET A$:REM ERWARTET TASTENDRUCK           <082>
521 IF A$="M" THEN POKE VIC+28,1:REM UMSCH
    ALTEN AUF MULTICOLORSPRITE                <116>
522 IF A$="H" THEN POKE VIC+28,0:REM UMSCH
    ALTEN AUF HIRES-SPRITE                    <217>
530 GOTO 340                                  <030>
600 REM ***** SPRITE DATEN *****              <131>
610 DATA 64,0,1,16,170,4,6,170,144,10,170,
    160,42,170,168,41,105,104,169,235,106     <165>
620 DATA 169,235,106,169,235,106,170,170,1
    70,170,170,170,170,170,170,170,170,170    <128>
630 DATA 166,170,154,169,85,106,170,85,170
    ,42,170,168,10,170,160,1,0,64,1,0,64      <206>
640 DATA 5,0,80,0                             <114>
```

Listing 14. Mit den Tasten <M> und <H> läßt sich das Sprite zwischen Hires- und Multicolor-Modus umschalten

Noch einmal: Sprites

Es gibt mindestens drei verschiedene Basic-Programmiertechniken zum Erstellen von Grafikbildern und Zeichentrickfilmen mit dem C64. Sie können den computereigenen Grafikzeichensatz benutzen. Sie können in Ihren eigenen Zeichen programmieren oder, die beste Möglichkeit... die in den Computer eingebauten Sprites benutzen.

Damit Sie sehen, wie einfach dies ist, zeigen wir Ihnen in Listing 15 ein kurzes Programm für die Erstellung von Sprites in Basic.

```
10 PRINT CHR$(147)                            <039>
20 POKE 2040,13                               <069>
30 FOR S=832 TO 832+62:POKE S,255:NEXT        <165>
40 V=53248                                    <173>
50 POKE V+21,1                                <197>
60 POKE V+39,1                                <241>
70 POKE V,24                                  <107>
80 POKE V+1,100                               <099>
```

Listing 15. Dieses kurze Programm beinhaltet schon das Wichtigste zur Erstellung von Sprites

Dieses Programm enthält die Hauptbestandteile, die Sie zum Erstellen von Sprites benötigen. Es definiert das erste Sprite – Sprite 0 – als weißes Quadrat auf dem Bildschirm. Wir wollen das Programm nun Zeile für Zeile erklären:

Zeile 10 löscht den Bildschirm.

Zeile 20 setzt den »Sprite-Zeiger« auf die Speicherstelle, aus der der C64 die Spritedaten lesen soll. Sprite 0 wird in 2040, Sprite 1 in 2041, Sprite 2 in 2042 usw. und Sprite 7 in 2047 gesetzt. Durch Verwendung der nachfolgenden Zeile anstelle von Zeile 20 können alle 8 Sprite-Zeiger auf 13 gesetzt werden:

```
FOR SP = 2040 TO 2047:POKE SP, 13:NEXT SP
```

Zeile 30 schreibt das erste Sprite (Sprite 0) in 63 Byte des Speichers beginnend bei Adresse 832 (jedes Sprite benötigt 63 Byte des Speichers). Das erste Sprite (Sprite 0) wird so in den Speicherplätzen 832 bis 894 abgelegt.

Zeile 40 setzt die Variable »V« gleich 53248, der Startadresse des Videochips. Durch diese Eingabe können wir die Formel (V + Zahl) für Spriteeingaben benutzen. Das ist deshalb sinnvoll, weil sie Speicherkapazität spart und das Arbeiten mit kleineren Zahlen ermöglicht. So haben wir

z. B. in Zeile 50 POKE V + 21 eingegeben. Das entspricht der Eingabe von POKE 53248 + 21 oder 53269. V+21, benötigt jedoch weniger Platz als 53269 und läßt sich leichter merken.

Zeile 50 aktiviert Sprite 0. Zum Einschalten der einzelnen Sprites oder einer Kombination von Sprites müssen Sie lediglich POKE V + 21, gefolgt von einer Zahl zwischen 0 (Ausschalten aller Sprites) und 255 (Einschalten aller 8 Sprites), eingeben. Durch das POKEn der Zahl in Tabelle 13 können ein oder mehrere Sprites eingeschaltet werden:

Alle Sprites	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Alle aus
V+21,255	V+21,1	V+21,2	V+21,4	V+21,8	V+21,16	V+21,32	V+21,64	V+21,128	V+21,0

Tabelle 13. Die POKEs zum Ein- und Ausschalten der Sprites

Durch POKE V+21,1 wird Sprite 0 eingeschaltet, POKE V+21,128 schaltet Sprite 7 ein. Auch Spritekombinationen sind erlaubt. So wird z. B. durch POKE V+21,129 sowohl Sprite 0 als auch Sprite 7 durch Addition der beiden Zahlen (1+128) eingeschaltet.

Zeile 60 legt die Farbe von Sprite 0 fest. Es gibt 16 mögliche Spritefarben, die von 0 (Schwarz) bis 15 (Grau) numeriert sind. Jedes Sprite benötigt für die Farbe eine unterschiedliche POKE-Anweisung von V+39 bis V+46. POKE V+39,1 gibt Sprite 0 die Farbe Weiß. Durch POKE V+46,15 erhält Sprite 7 die Farbe Grau.

Beim Erstellen eines Sprite bleibt dieses so lange im Speicher erhalten, bis es neu definiert oder der Computer abgeschaltet wird. Auf diese Weise können Farbe, Position und Form des Sprite im Direktmodus geändert werden. Dies ist besonders sinnvoll beim Editieren.

Führen Sie z. B. obiges Programm aus, und geben Sie danach diese Zeile im Direktmodus (ohne Zeilennummer) ein. Danach drücken Sie <RETURN>.

```
POKE V+39,8
```

Das Sprite auf dem Bildschirm ist nun orange. Versuchen Sie das POKEn einer anderen Zahl zwischen 0 und 15. Es wird eine andere Spritefarbe erscheinen. Da dies im Direktmodus erfolgte, wird das Sprite beim Ausführen des Programms wieder die ursprüngliche Farbe (Weiß) haben.

Zeile 70 bestimmt die horizontale oder »X«-Position des Sprite auf dem Bildschirm. Diese Zahl legt die Position der oberen linken Ecke des Sprite fest. Das Sprite berührt den linken Bildschirmrand, wenn es in Position 24 steht. Mit kleineren Werten kann es links aus dem Bildschirm heraus bewegt werden.

Zeile 80 bestimmt die vertikale oder »Y«-Position des Sprite. In diesem Programm plazierten wir das Sprite an die »X«-Position 100. Probieren Sie eine andere Position aus. Geben Sie dazu folgende POKE-Anweisung im Direktmodus ein und drücken Sie danach <RETURN>.

```
POKE V,24:POKE V+1,50
```

Hierdurch wird das Sprite in die obere linke Bildschirmecke gesetzt. Um das Sprite in die untere linke Ecke zu bewegen, geben Sie ein:

```
POKE V,24:POKE V+1,229
```

Jede Adresse von 832 bis 895, also der Speicherbereich von Sprite 0, repräsentiert einen Satz von 8 Pixel, wobei drei 8-Pixel-Sätze eine horizontale Reihe des Sprite darstellen. Die Schleife in Zeile 30 gibt dem Computer die Anweisung POKE 832,255, wodurch die ersten 8 Pixel »ausgefüllt« werden, danach werden durch POKE 833,255 die nächsten 8 Pixel ebenfalls »ausgefüllt« usw., bis zu Adresse 894, die die letzte Gruppe von 8 Pixel in der unteren rechten Spriteecke angibt.

Damit Sie besser sehen, wie dies funktioniert, versuchen Sie folgendes im Direktmodus und beachten Sie, daß die zweite Gruppe der 8 Pixel gelöscht wird:

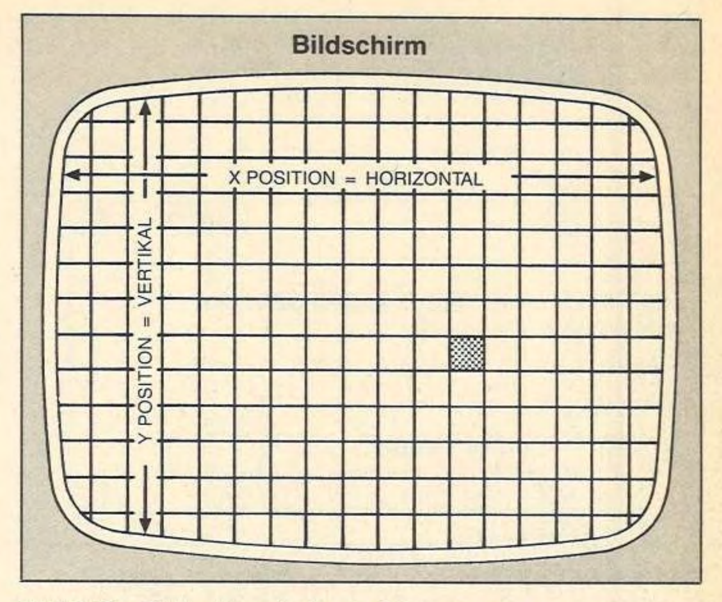


Bild 8. Der Bildschirm ist in ein Gitter aus X- und Y-Koordinaten unterteilt

```
POKE 833,0
POKE 833,255
```

Durch folgende Zeile, die Sie in Ihr Programm aufnehmen können, wird die Mitte des Sprite gelöscht:

```
90 FOR A = 836 TO 891 STEP 3:POKE A,0:NEXT A
```

Denken Sie daran, daß die Pixel, aus denen die Sprites aufgebaut sind, in Sätzen von jeweils 8 (1 Byte) gruppiert sind. Diese Zeile löscht die fünfte Gruppe von 8 Pixel (Satz 836) und jeden dritten Satz bis zu Satz 890. Versuchen Sie, andere Zahlen in die Adressen von 832 bis 894 zu POKEn.

Positionierung der Sprites auf dem Bildschirm

Der gesamte Bildschirm ist wie ein Koordinatensystem in X- und Y-Koordinaten unterteilt. Die X-Koordinate ist die horizontale Position und die Y-Koordinate die vertikale Position auf dem Bildschirm (siehe Bild 8).

Um die Sprites auf dem Bildschirm zu positionieren, müssen zwei Eingaben – X- und Y-Position – gePOKEt werden. Auf diese Weise erfährt der Computer, wo sich die obere linke Ecke des Sprite auf dem Bildschirm befinden soll. Bitte denken Sie daran, daß ein Sprite aus 504 einzelnen Pixel (24 horizontal mal 21 vertikal) besteht. Die Anzeige beginnt in der von Ihnen definierten X-Y-Position. Sie basiert stets auf der oberen linken Ecke, auch wenn Sie für das gesamte Sprite lediglich einen kleinen Teil des 24 x 21-Pixel-Spritebereichs benutzen. Die Funktionsweise der X-Y-Positionierung können Sie Bild 9 entnehmen. Dieses zeigt die X- und Y-Werte in Zusammenhang mit der Bildschirmanzeige. Bitte beachten Sie, daß der weiße Bereich den Teil außerhalb des Bildschirms angibt.

Um ein Sprite an der gewählten Stelle anzuzeigen, müssen Sie die X- und Y-Eingabe für jedes Sprite POKEn...denken Sie daran, daß jedes Sprite seine eigene X- und Y-POKE-Anweisung hat. In Tabelle 14 sehen Sie die X- und Y-Eingaben für alle 8 Sprites.

V=53248	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite7
SET X	V,X	V+2,X	V+4,X	V+6,X	V+8,X	V+10,X	V+12,X	V+14,X
SET Y	V+1,Y	V+3,Y	V+5,Y	V+7,Y	V+9,Y	V+11,Y	V+13,Y	V+15,Y
MSB-X	V+16,1	V+16,2	V+16,4	V+16,8	V+16,16	V+16,32	V+16,64	V+16,128

Tabelle 14. Zum Setzen der X-Y-Sprite-Positionen verwenden Sie bitte diese Werte

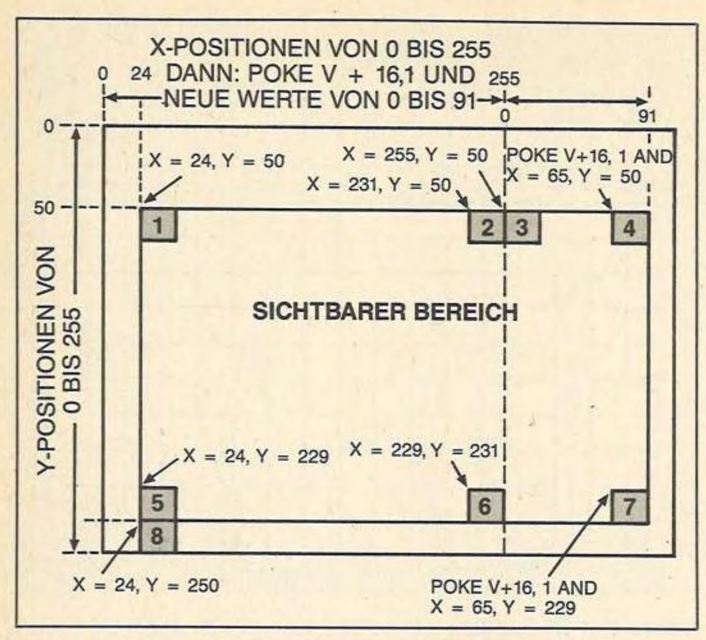


Bild 9. Bestimmung der X-/ Y-Sprite-Position

Setzen einer X-Position:

Die möglichen X-Werte sind, gezählt von links nach rechts, 0 bis 255. Die Werte 0 bis 23 plazieren alles oder einen Teil des Sprite außerhalb des sichtbaren Bereichs auf der linken Bildschirmseite - die Werte 24 bis 255 zeigen das Sprite im sichtbaren Bereich bis zur 255. Position an. Um ein Sprite in eine dieser Positionen zu plazieren, geben Sie lediglich die X-Positions-POKE-Anweisung für das benutzte Sprite ein. Um z. B. Sprite 1 an die äußerste linke X-Position im sichtbaren Bereich zu POKEn, geben Sie folgendes ein: POKE V + 2,24.

X-Werte oberhalb der 255. Position:

Um über die 255. Position des Bildschirms hinauszugelangen, benötigen Sie eine zweite POKE-Anweisung. Normalerweise geht die horizontale Numerierung (X) über die 255. Position hinaus. Da die Register jedoch nur 8 Bit enthalten, müssen wir ein »zweites Register« erstellen, um auf die rechte Bildschirmseite zu gelangen. Die X-Numerierung beginnt hier wieder mit 0. Um also über die X-Position 255 hinauszukommen, ist POKE V+16 sowie eine Zahl (abhängig vom Sprite, siehe Tabelle 14) erforderlich. Hierdurch erhalten Sie 64 zusätzliche X-Positionen (numeriert von 0 bis 65) im sichtbaren Bereich auf der rechten Bildschirmseite.

V=53248	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite7
SET X SET Y MSB-X	V,X V+1,Y V+16,1	V+2,X V+3,Y V+16,2	V+4,X V+5,Y V+16,4	V+6,X V+7,Y V+16,8	V+8,X V+9,Y V+16,16	V+10,X V+11,Y V+16,32	V+12,X V+13,Y V+16,64	V+14,X V+15,Y V+16,128

Tabelle 14. Zum Setzen der X-Y-Sprite-Positionen verwenden Sie bitte diese Werte

Setzen einer Y-Position:

Die möglichen Y-Werte sind 0 bis 255 und werden von oben nach unten gezählt. Durch die Werte 0 bis 49 wird das Sprite ganz oder teilweise außerhalb des sichtbaren Bereichs oben am Bildschirm gezeigt. Mit den Werten 50 bis 259 befindet sich das Sprite im sichtbaren Bereich. Durch die Werte 230 bis 255 wird das Sprite ganz oder teilweise aus dem sichtbaren Bereich hinausbewegt.

Wir wollen uns nun mit der X-Y-Positionierung beschäftigen und nehmen hierzu Sprite 1 als Beispiel. Geben Sie hierzu Listing 16 ein.

Dieses einfache Programm zeigt Sprite 1 als Kästchen

```
10 PRINT CHR$(147):V=53248:POKE V+21,2:POK
   E 2041,13:FOR S=832 TO 895:POKE S,255:N
   EXT                                  <127>
20 POKE V+40,7                          <173>
30 POKE V+2,24                          <118>
40 POKE V+3,50                          <029>
```

Listing 16. Ein Beispiel zur Sprite-Positionierung

an und setzt es in die obere linke Bildschirmecke. Ändern Sie die Zeile 40 nun wie folgt:

```
40 POKE V+3,229
```

Hierdurch wird das Sprite in die untere linke Bildschirmecke bewegt. Nun wollen wir die rechte X-Grenze des Sprite überprüfen: Ändern Sie Zeile 30 wie folgt:

```
30 POKE V+2,255
```

Hierdurch wird das Sprite nach rechts bewegt. Es erreicht die rechte X-Grenze, die durch 255 festgelegt ist. Von diesem Punkt an muß das höchste Bit in Register 16 gesetzt werden. Sie müssen POKE V+16 sowie eine Zahl eingeben, die in der rechten »X«-Spalte in der X-Y-POKE-Tabelle (siehe Bild 9) angezeigt wird. Auf diese Weise wird der X-Positionszähler bei der 256. Pixel-Position auf dem Bildschirm neu gestartet. Ändern Sie Zeile 30 wie folgt:

```
30 POKE V+16, PEEK(V+16) OR 2:POKE V+2,0
```

Durch POKE V +16,2 wird das höchste Bit der X-Position für Sprite 1 gesetzt und bei der 256. Pixel-Position auf dem Bildschirm ein neuer 0-Punkt festgelegt. Durch POKE V+2,0 wird das Sprite an der neuen 0-Position, die nun auf den 256. Pixel gesetzt ist, abgebildet.

Um zurück zur linken Bildschirmseite zu gelangen, muß das höchste Bit des X-Positionszählers wieder auf 0 gesetzt werden. Geben Sie hierzu für Sprite 1 folgendes ein:

```
POKE V+16, PEEK (V+16) AND 253
```

Positionieren mehrerer Sprites auf dem Bildschirm

In Listing 17 sehen Sie ein Programm, das drei verschiedene Sprites (0,1 und 2) in verschiedenen Farben definiert und in verschiedenen Positionen auf dem Bildschirm darstellt.

```
10 PRINT CHR$(147):V=53248:FOR S=832 TO 89
   5:POKE S,255:NEXT                    <212>
20 FOR M=2040 TO 2042:POKE M,13:NEXT    <180>
30 POKE V+21,7                          <183>
40 POKE V+39,1:POKE V+40,7:POKE V+41,8  <056>
50 POKE V,24:POKE V+1,50                <094>
60 POKE V+2,12:POKE V+3,229             <109>
70 POKE V+4,255:POKE V+5,50             <159>
```

Listing 17. Bringt mehrere Sprites in verschiedenen Farben auf den Bildschirm

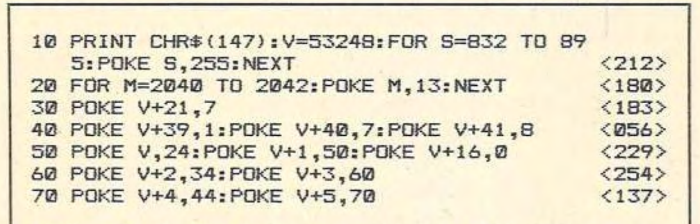
Der Einfachheit halber sind alle drei Sprites als durchgehende Quadrate definiert, die ihre Daten alle aus demselben Speicherbereich erhalten. Wichtig ist hierbei, wie alle drei Sprites positioniert werden. Das weiße Sprite 0 befindet sich in der oberen linken Ecke, das gelbe Sprite 1 in der unteren Ecke, jedoch halb außerhalb des Bildschirms. Denken Sie daran, 24 ist die äußerste linke X-Position im sichtbaren Bereich... durch eine X-Position unter 24 wird das Sprite ganz oder teilweise aus dem Bildschirm »hinausgeschoben«. Wir haben hier die X-Position 12 benutzt, so daß die Hälfte des Sprite außerhalb des Bildschirms liegt. Das orangefarbene Sprite 2 liegt an der rechten X-Grenze (Position 255).

Spriteprioritäten

Verschiedene Sprites können sich vor- bzw. hintereinander auf dem Bildschirm bewegen. Dieser dreidimensionale Effekt wird durch die Spriteprioritäten erzielt, die bestim-

men, welches Sprite bei einer eventuellen Überdeckung auf dem Bildschirm Vorrang gegenüber dem anderen hat.

Die Regelung »wer zuerst kommt, mahlt zuerst«, gilt auch hier: Das Sprite mit der niedrigeren Zahl hat automatisch Priorität über Sprites mit höheren Zahlen. Werden z. B. Sprite 0 und Sprite 1 an der gleichen Stelle des Bildschirms angezeigt, so erscheint Sprite 0 vor Sprite 1. Sprite 0 hat daher stets Vorrang vor allen anderen Sprites, da es dasjenige mit der niedrigsten Zahl ist. Sprite 1 hat Priorität über die Sprites 2 bis 7, Sprite 2 Vorrang vor den Sprites 3 bis 7 usw. Sprite 7 (das letzte Sprite) hat die niedrigste Priorität und wird daher bei einer Überdeckung stets hinter allen anderen Sprites erscheinen.



```
10 PRINT CHR$(147):V=53248:FOR S=832 TO 89
   5:POKE S,255:NEXT                          <212>
20 FOR M=2040 TO 2042:POKE M,13:NEXT          <180>
30 POKE V+21,7                                <183>
40 POKE V+39,1:POKE V+40,7:POKE V+41,8        <056>
50 POKE V,24:POKE V+1,50:POKE V+16,0          <229>
60 POKE V+2,34:POKE V+3,60                    <254>
70 POKE V+4,44:POKE V+5,70                    <137>
```

Listing 18. Sprites haben verschiedene Prioritäten

Ändern Sie die Zeilen 50, 60 und 70 des letzten Programms wie in Listing 18 dargestellt.

Sie müssen nun ein weißes Sprite über einem gelben Sprite sehen und unter diesen beiden muß ein orangefarbenes Sprite angezeigt sein. Nun wissen Sie, wie die Prioritäten gesetzt sind und können die Sprites beliebig bewegen. Dies hilft bei der Programmierung von Trickfilmen.

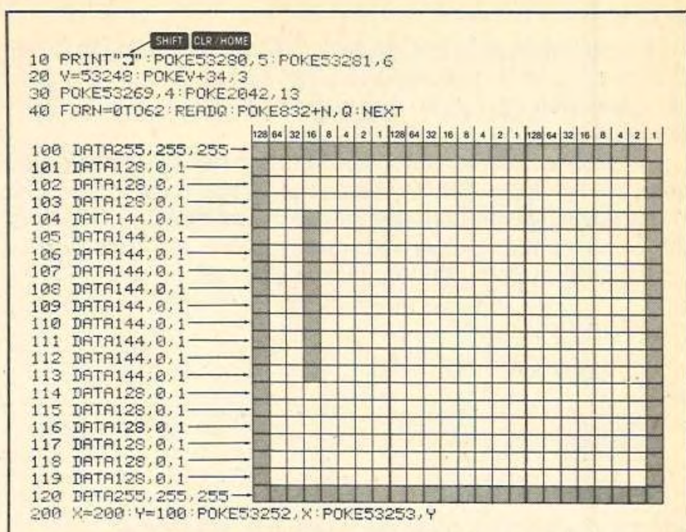
Zeichnen eines Sprite

Das Zeichnen eines Sprite verläuft genauso wie das Ausmalen eines Malbuchs. Jedes Sprite besteht aus winzigen Punkten, die Pixel genannt werden. Um ein Sprite zu zeichnen, brauchen Sie lediglich einige der Pixel »auszumalen«.

Sehen Sie sich das Gitter in Bild 10 an. So sieht ein leeres Sprite aus.

1 1 1
2 6 3 1 2 2 3 1 2 6 3 1
8 4 2 6 8 4 2 1 8 4 2 6 8 4 2 1 8 4 2 6 8 4 2 1

Bild 10. Ein Gitter für die Sprite-Erstellung



```
10 PRINT"[SHIFT CLR/HOME]":POKE53280,5:POKE53281,6
20 V=53248:POKEV+34,3
30 POKE53269,4:POKE2042,13
40 FORN=0TO62:READQ:POKE832+N,Q:NEXT
100 DATA255,255,255
101 DATA128,0,1
102 DATA128,0,1
103 DATA128,0,1
104 DATA144,0,1
105 DATA144,0,1
106 DATA144,0,1
107 DATA144,0,1
108 DATA144,0,1
109 DATA144,0,1
110 DATA144,0,1
111 DATA144,0,1
112 DATA144,0,1
113 DATA144,0,1
114 DATA128,0,1
115 DATA128,0,1
116 DATA128,0,1
117 DATA128,0,1
118 DATA128,0,1
119 DATA128,0,1
120 DATA255,255,255
200 X=200:Y=100:POKE53252,X:POKE53253,Y
```


Listing 19. Ein anschauliches Sprite-Entwurfsprogramm mit nebenliegendem Entwurfsblatt

Jedes kleine »Kästchen« stellt ein Pixel im Sprite dar. Es gibt 24 horizontale mal 21 vertikale oder insgesamt 504 Pixel pro Sprite. Um dem Sprite nun eine bestimmte Form zu geben, müssen Sie diese Pixel im einem Programm ausmalen.... Wie können jedoch mehr als 500 Pixel angesteuert werden? Sie müssen nicht 504 einzelne Zahlen, sondern lediglich 63 Zahlen für jedes Sprite eingeben. Das funktioniert folgendermaßen: Um die Erstellung von Sprites für Sie so einfach wie möglich darzustellen, wollen wir sie schrittweise erklären:

Schritt 1:

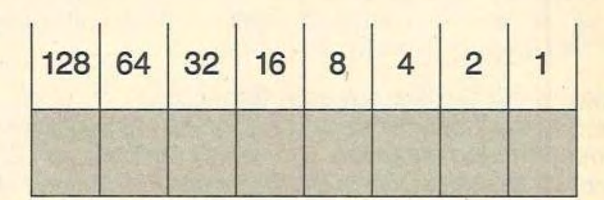
Schreiben Sie das Sprite-Erstellungs-Programm, wie in Listing 19 gezeigt, auf ein Stück Papier. Bitte beachten Sie, daß in Zeile 100 ein Abschnitt mit DATAs beginnt, der die 63 Zahlen für die Spriteerstellung enthält.

Schritt 2:

Malen Sie die Pixel im Gitter in Bild 10 aus (oder nehmen Sie ein Blatt Millimeterpapier. Denken Sie daran, daß ein Sprite aus 24 horizontalen mal 21 vertikalen Kästchen besteht). Benutzen Sie einen Bleistift und drücken Sie nicht zu fest auf, damit Sie dieses Gitter wieder benutzen können (oder machen Sie sich einige Fotokopien des Gitters). Sie können beliebige Bilder erstellen. Am Anfang wollen wir jedoch als Beispiel einen einfachen Kasten zeichnen.

Schritt 3:

Sehen Sie sich die ersten acht Pixel an. Über jeder Spalte von Pixel steht eine Zahl (128, 64, 32, 16, 8, 4, 2). Die besondere Art der Addition, die wir benutzen wollen, stammt aus der Binär-Arithmetik, die bei Computern oft verwendet wird. Bild 11 zeigt Ihnen die ersten acht Pixel in der oberen linken Spriteecke.



128	64	32	16	8	4	2	1

Bild 11. Die ersten 8 Pixel der oberen linken Ecke des Sprite

Schritt 4:

Addieren Sie die Zahlen der ersten ausgemalten Pixel. Die erste Gruppe der acht Pixel ist vollständig ausgemalt, so daß sich eine Summe von 255 ergibt.

Schritt 5:
Geben Sie diese Zahl als erste DATA-Anweisung in Zeile 100 des Sprite-Erstellungsprogramms ein. Geben Sie 255 für die zweite und dritte Achtergruppe ein.
Schritt 6:
Sehen Sie sich die ersten acht Pixel in der zweiten Spritereihe an. Addieren Sie die Werte der ausgemalten Pixel. Da in unserem Beispiel nur ein Pixel ausgemalt ist, ergibt sich die Summe 128. Geben Sie diesen Wert als erste DATA-Anweisung in Zeile 101 ein (siehe Bild 12).

128	64	32	16	8	4	2	1

Bild 12. Die ersten 8 Pixel der zweiten Sprite-Reihe

Schritt 7:
Addieren Sie die Werte der nächsten Gruppe von acht Pixel (die Summe ist 0, da hier alle Pixel leer sind). Geben Sie diese Zahl in Zeile 101 ein. Nun nehmen wir uns die nächste Gruppe vor und führen das gleiche für alle Achtergruppen durch (es gibt drei Gruppen pro Reihe und insgesamt 21 Reihen). Es ergibt sich also eine Gesamtanzahl von 63. Jede Zahl gibt eine Gruppe zu je acht Pixel an, und 63 Gruppen mit acht Pixel ergeben insgesamt 504 vollständig unabhängige Pixel. Das Programm läßt sich vielleicht noch besser wie folgt erklären: Jede Programmzeile stellt eine Reihe im Sprite dar. Jede der drei Zahlen in jeder Reihe steht für eine Gruppe mit je acht Pixel. Und jede Zahl weist den Computer an, welches Pixel ausgemalt und welches leer sein soll.
Schritt 8:
Komprimieren Sie das Programm. Dazu werden alle DATA-Anweisungen wie in Listing 20 zusammengefaßt. Beachten Sie, daß Sie das Spriteprogramm zunächst auf ein

```
10 PRINT CHR$(147):POKE 53280,5:POKE 53281
   ,6                                         <061>
20 V=53248:POKE V+34,3                        <131>
30 POKE 53269,4:POKE 2042,13                  <168>
40 FOR N=0 TO 62:READ Q:POKE 832+N,Q:NEXT     <188>
100 DATA 255,255,255,128,0,1,128,0,1,128,0
    ,1,144,0,1,144,0,1,144,0,1,144,0,1        <115>
101 DATA 144,0,1,144,0,1,144,0,1,144,0,1,1
    44,0,1,144,0,1,128,0,1,128,0,1            <029>
102 DATA 128,0,1,128,0,1,128,0,1,128,0,1,2
    55,255,255                                <075>
200 X=200:Y=100:POKE 53252,X:POKE 53253,Y     <024>
```

Listing 20. Das Entwurfsprogramm in komprimierter Form

Blatt Papier schreiben sollten. Das hat einen guten Grund. Die DATA-Anweisungszeilen 100 bis 120 im Programm in Schritt 1 sollen Ihnen lediglich zeigen, welche Zahl zu welcher Pixelgruppe Ihres Sprite gehört.

Bewegen der Sprites auf dem Bildschirm

Jetzt ist das Sprite fertig, und wir können es nun zu interessanten Dingen benutzen. Um das Sprite über den Bildschirm zu bewegen, fügen Sie folgende zwei Zeilen in Ihr Programm ein:

```
50 POKE V+5,100:FOR X=24 TO 255:POKE V+4,X:NEXT:
   POKE V +16,4
55 FOR X = 0 TO 65:POKE V+4, X:NEXT X:POKE V+
   16,0 : GOTO 50
```

Durch Zeile 50 wird die Y-Position bei 100 gesetzt (probieren Sie auch 50 oder 229 aus). Dann wird eine FOR-NEXT-Schleife aufgebaut, durch die das Sprite nacheinander in die X-Positionen 0 bis 255 gesetzt wird. Beim Erreichen der 255. Position wird das MSB gesetzt (POKE V +16,2), das zum Erreichen des rechten Bildschirmrandes benötigt wird.

Zeile 55 enthält ebenfalls eine FOR-NEXT-Schleife, durch die das Sprite in die letzten 65 Bildschirmpositionen gebracht wird. Bitte beachten Sie, daß der X-Wert auf 0 zurückgestellt wurde. Da Sie jedoch das höchste Bit der X-Position gesetzt hatten (POKE V + 16,2), beginnt X auf der rechten Bildschirmseite ab Position 256.

Diese Zeile wird immer wieder durchlaufen (GOTO 50). Soll das Sprite sich nur einmal über den Bildschirm bewegen und dann verschwinden, nehmen Sie GOTO 50 einfach heraus.

Nachstehend sehen Sie eine Zeile, durch die das Sprite vor- und zurückbewegt wird:

```
50 POKE V+5,100:FOR X=24 TO 255:POKE V +4,X: NEXT:
   POKE V+16,4:FOR X= 0 TO 65: POKE V+4,X: NEXT X
55 FOR X = 65 TO 0 STEP-1:POKE V+4,X:NEXT:POKE V+
   16,0:FOR X=255 TO 24 STEP-1:POKE V+4,X:NEXT
60 GOTO 50
```

Sehen Sie, wie dieses Programm funktioniert? Es ist das gleiche wie das vorherige. Nur wird hier beim Erreichen der rechten Bildschirmseite der Ablauf stets umgekehrt, so daß das Sprite sich wieder in die andere Richtung bewegt. Dies wird durch STEP-1 bewirkt. Das Programm wird angewiesen, das Sprite in die X-Werte von 65 bis 0 auf der rechten

```
5 S=54272:POKE S+24,15:POKE S,220:POKE S+1
  ,68:POKE S+5,15:POKE 6+6,215                <070>
10 POKE S+7,120:POKE S+8,100:POKE S+12,15:
   POKE S+13,215                              <166>
15 PRINT CHR$(147):V=53248:POKE V+21,1        <044>
20 FOR S1=12288 TO 12350:READ Q1:POKE S1,Q
   1:NEXT                                     <102>
25 FOR S2=12352 TO 12414:READ Q2:POKE S2,Q
   2:NEXT                                     <186>
30 FOR S3=12416 TO 12478:READ Q3:POKE S3,Q
   3:NEXT                                     <248>
35 POKE V+39,15:POKE V+1,68                   <168>
40 PRINT TAB(160)"{WHITE}ICH BIN DIE TANZM
   AUS !{LIG.BLUE}"                           <168>
45 P=192                                      <201>
50 FOR X=0 TO 347 STEP 3                      <206>
55 RX=INT(X/256):LX=X-RX*256                  <031>
60 POKE V,LX:POKE V+16,RX                     <232>
70 IF P=192 THEN GOSUB 200                    <188>
75 IF P=193 THEN GOSUB 300                    <069>
80 POKE 2040,P:FOR T=1 TO 60:NEXT             <153>
85 P=P+1:IF P>194 THEN P=192                  <027>
90 NEXT                                       <100>
95 END                                        <097>
100 DATA 30,0,120,63,0,252,127,129,254,127
    ,129,254,127,189,254,127,255,254          <197>
101 DATA 63,255,252,31,187,248,3,187,192,1
    ,255,128,3,189,192,1,231,128,1,255,0      <161>
102 DATA 31,255,0,0,124,0,0,254,0,1,199,32
    ,3,131,224,7,1,192,1,192,0,3,192,0        <073>
103 DATA 30,0,120,63,0,252,127,129,254,127
    ,129,254,127,189,254,127,255,254          <200>
104 DATA 63,255,252,31,221,248,3,221,192,1
    ,255,128,3,255,192,1,195,128,1,231,3      <211>
105 DATA 31,255,255,0,124,0,0,254,0,1,199,
    0,7,1,128,7,0,204,1,128,124,7,128,56      <170>
106 DATA 30,0,120,63,0,252,127,129,254,127
    ,129,254,127,189,254,127,255,254          <203>
107 DATA 63,255,252,31,221,248,3,221,192,1
    ,255,134,3,189,204,1,199,152,1,255,48     <051>
108 DATA 1,255,224,1,252,0,3,254,0            <219>
109 DATA 7,14,0,204,14,0,248,56,0,112,112,
    0,0,60,0,-1                               <100>
200 POKE S+4,129:POKE S+4,128:RETURN          <182>
300 POKE S+11,129:POKE S+11,128:RETURN        <018>
```

Listing 21. »Die Tanzmaus«, ein Beispiel für Trickfilmeffekte mit Sprites

Bildschirmseite und darüber hinaus von 255 bis 0 auf der linken Bildschirmseite zu positionieren. Hierzu wird jeweils um den Schritt -1 zurückgegangen.

Vertikales Rollen

Diese Art der Spritebewegung wird »Rollen« genannt. Um das Sprite auf diese Weise nach oben oder unten in die Y-Position zu bewegen, brauchen Sie lediglich eine Zeile. Löschen Sie die Zeilen 50 und 55, indem Sie die Zeilennummern eingeben und danach jeweils <RETURN< drücken.

Geben Sie nun Zeile 50 wie folgt ein:

```
50 POKE V+4,24:FOR Y = 0 TO 255:POKE V+5,Y:NEXT
```

Die Tanzmaus – ein Sprite-Programmbeispiel

Gelegentlich sind die in Programmieranleitungen beschriebenen Techniken nur schwer zu verstehen. Aus diesem Grund haben wir in Listing 21 ein Spriteprogramm erstellt, das drei verschiedene Sprites in einem Zeichentrick mit Geräuscheffekten benutzt. Damit Sie genau verstehen, wie dieses Programm funktioniert, haben wir jeden Befehl erklärt:

Zeile 5:

S=54272	Setzt die Variable »S« gleich 54272, also der Anfangsspeicheradresse des Sound-Chip. Statt nun einen direkten Speicherplatz zu POKEn, werden wir ab jetzt POKE S+X eingeben.
POKE S+24,15	Entspricht POKE 54296,15. Hierdurch wird die höchste Lautstärke eingestellt.
POKE S,220	Entspricht POKE 54272,220. Setzt das Low-Byte in Stimme 1 für eine Note, die ungefähr dem hohen C in Oktave 6 entspricht.
POKE S+1,68	Entspricht POKE 54277,15. Setzt das Attack/Decay für Stimme 1 und besteht in diesem Fall aus dem maximalen Abklingpegel ohne Einsetzen. Hierdurch entsteht der Echo-Effekt.
POKE S+6,215	Entspricht POKE 54278,215. Setzt das Sustain/Release für Stimme 1 (215 stellt eine Kombination zwischen Sustain- und Releasezeit dar.)

Zeile 10:

POKE S+7,120	Entspricht POKE 54279,120. Setzt Low-Byte für Stimme 2.
POKE S+8,100	Entspricht POKE 54280,100. Setzt High-Byte für Stimme 2.
POKE S+12,15	Entspricht POKE 54284,15. Setzt Attack/Decay für Stimme 2 auf den gleichen Pegel wie für Stimme 1.
POKE S+13,215	Entspricht POKE 54285,215. Setzt das Sustain/Release für Stimme 2 auf den gleichen Pegel wie für Stimme 1.

Zeile 15

PRINT CHR$(147)	Löscht den Bildschirm bei Programmbeginn.
V = 53248	Definiert die Variable »V« als Startadresse des VIC-Chip, der die Sprites steuert. Von nun an werden alle Sprite-Register des VIC als V plus einen Wert definiert.
POKE V+21.1	Schaltet die Spritenummer 1 ein (Aktivierung).

Zeile 20:

FOR S1=12288 TO 12350	In diesem Zeichentrick benutzen wir ein Sprite (Sprite 0). Wir werden jedoch drei verschiedene Spritedaten für die Definition von drei unterschiedlichen Formen benutzen. Dazu schalten wir die Zeiger für Sprite 0 auf drei verschiedene Speicherplätze, in denen die Daten für die Definition der unterschiedlichen Formen gespeichert sind. Dasselbe Sprite wird hintereinander schnell in drei verschiedenen Formen definiert. Hierdurch entsteht der Tanzmaus-Trickfilm. Sie können Dutzende von Spriteformen in DATA-Anweisungen benutzen und diese Formen mit einem oder mehreren Sprites ansteuern. Sie brauchen daher nicht ein Sprite auf eine Form zu begrenzen (und umgekehrt). Ein Sprite kann viele verschiedene Formen haben, indem einfach die Pointer für dieses Sprite auf verschiedene Adressen zeigen. In den Speicherplätzen sind dann die Spritedaten der verschiedenen Formen gespeichert. Diese Zeile bedeutet, daß wir die Daten für »Spriteform 1« in die Speicherplätze 12288 bis 12350 eingegeben haben.
READ Q1	Liest nacheinander 63 Zahlen der DATA-Anweisung, beginnend bei Zeile 100. Q1 ist ein beliebiger Variablenname.
POKE S1,Q1	POKEt die erste Zahl der DATA-Anweisungen (erstes Q1 ist 30) in den ersten Speicherplatz (12288). Entspricht POKE 12288,30.
NEXT	Weist den Computer an, die Befehle zwischen den Teilen FOR und NEXT der Schleife auszuführen. (READ Q1 und POKE S1,Q1 mit den NEXT-Zahlen). Durch die NEXT-Anweisung liest der Computer NEXT Q1 von den DATA-Anweisungen. NEXT Q1 ist 0. Außerdem wird S1 um 1 erhöht, dies entspricht 12289. Das Ergebnis ist POKE 12289,0... durch den NEXT-Befehl wird die Schleife bis zu den letzten Werten der Serie durchgeführt, also bis zu POKE 12350,0.

Zeile 25:

FOR S2=12352 TO 12414	Die zweite Form von Sprite 0 wird durch die DATAs definiert, die in die Adressen 12352 bis 12414 geschrieben werden. Bitte beachten Sie, daß Adresse 12351 übersprungen wird. Dies ist der 64. Platz der Definition der ersten Sprite-Gruppe. Er enthält jedoch keine Sprite-Daten. Beachten Sie bei der Sprite-Definition, daß 64 Plätze benutzt werden. Sprite-Daten werden jedoch nur in die ersten 63 Plätze gePOKEt.
READ Q2	Durch diese READ-Anweisung wird die nächste Zahl im DATA-Bereich gesucht, und 63 Zahlen werden nacheinander gelesen.
POKE S2,Q2	Hierdurch werden die Daten (Q2) in die Speicherplätze (S2) für unsere zweite Sprite-Form gePOKEt, die bei Adresse 12352 beginnt.
NEXT	Entspricht Zeile 20.

Zeile 30:

FOR S3=12416 TO 12478	Die dritte Form von Sprite 0 wird durch die DATAs in den Adressen 12416 bis 12478 definiert.
READ Q3	Liest nacheinander die letzten 63 Zahlen als Q3.
POKE S3,Q3	POKEt diese Zahlen in die Adressen 12416 bis 12478
NEXT	Entspricht den Zeilen 20 und 25.

Zeile 35:

POKE V+39,15	Setzt für Sprite 0 die Farbe Hellgrau.
POKE V+1,68	Setzt die obere rechte Ecke des Sprite-Quadrats in die Vertikalposition 68 (Y). Zum Vergleich: Position 50 ist die oberste Position auf dem sichtbaren Bildschirm.

Zeile 40:

PRINT TAB(160)	Hierdurch wird der Cursor um 160 Leerstellen ab der oberen linken Bildschirmecke versetzt – dies entspricht vier Reihen. Hierdurch beginnt die PRINT-Meldung in der fünften Zeile auf dem Bildschirm.
{WHITE}	Die Tasten <CTRL> und <2> gleichzeitig drücken. Geschieht dies innerhalb von Anführungszeichen, so erscheint ein »reverses E«. Hierdurch wird die Farbe aller nachfolgenden Eingaben Weiß.
ICH BIN DIE TANZMAUS!	Dies ist eine einfache PRINT-Anweisung.
{LIG.BLUE}	Hierdurch wird die Farbe nach Ende der PRINT-Anweisung von Schwarz auf Hellblau geändert. Durch gleichzeitiges Drücken der Tasten <CBM> und <7> innerhalb von Anführungszeichen wird eine »negativ dargestellte Raute« angezeigt.

Zeile 45:

P=192	Setzt die Variable P gleich 192. Die Zahl 192 ist der zu benutzende Zeiger. In diesem Fall wird Sprite 0 aus den Speicherplätzen ausgelesen, die ab Adresse 12288 beginnen. Durch »Verstellen« des Zeigers auf die Adressen der beiden anderen Spriteformen kann mit nur einem Sprite ein Trickfilm mit drei verschiedenen Formen erstellt werden.

Zeile 50:

FOR X=0 TO	Bewegt das Sprite von Position 0 bis Position 347 in

347 STEP 3	3er-Schritten (hierdurch entsteht schnelle Bewegung).

Zeile 55:

RX=INT(X/256)	RX ist der ganzzahlige Anteil von X/256, was bedeutet, daß RX auf 0 gerundet wird, wenn X kleiner als 256 ist, und auf 1, wenn X die Position 256 erreicht. Wir werden RX gleich für die Anweisung POKE V + 16 mit einer 0 oder 1 benutzen, um die rechte Bildschirmseite »einzuschalten«.
LX=X-RX*256	Wenn sich das Sprite an der X-Position 0 befindet, sieht die Gleichung wie folgt aus: LX=0- (0 mal 256) =0. Wenn sich das Sprite an der X-Position 1 befindet, sieht die Gleichung wie folgt aus: LX=1 -(0 mal 256) =1. Wenn sich das Sprite an der X-Position 256 befindet, sieht die Gleichung so aus: LX=256 -(1 mal 256) =0. Hierdurch wird X zurück auf 0 gesetzt. Dies ist erforderlich, wenn die Bewegung bis zum rechten Rand erfolgen soll (POKE V + 16,1).

Zeile 60:

POKE V,LX	Mit der Anweisung POKE V wird die horizontale Position (X) von Sprite 0 auf den Bildschirm gesteuert (siehe Tabelle 15). Wie oben gezeigt, ändert sich der Wert von LX (horizontale Spriteposition) von 0 bis 255. Wenn er 255 erreicht, wird er automatisch aufgrund der Gleichung für LX in Zeile 55 auf 0 zurückgestellt.
POKE V+16,RX	Durch POKE V+16 wird stets die rechte Bildschirmseite nach Erreichen der Position 256 eingeschaltet, um die horizontalen Positionierungskoordinaten auf 0 zurückzustellen. RX ist entweder 0 oder 1, je nach der durch die Gleichung RX in Zeile 55 bestimmten Spriteposition.

Zeile 70:

IF P=192 THEN GOSUB 200	Ist der Sprite-Pointer auf 192 gesetzt (erste Spriteform), dann wird die Wellenformsteuerung des ersten Geräuscheffekts in Zeile 200 auf 129 und 128 gesetzt.

Zeile 75:

IF P=193 THEN GOSUB 300	Ist der Sprite-Pointer auf 193 (zweite Spriteform) gesetzt, dann wird die Wellenformsteuerung für den zweiten Geräuscheffekt (Stimme 2) auf 129 und 128 in Zeile 300 gesetzt.

Zeile 80:

POKE 2040,P	Setzt den Sprite-Pointer auf Adresse 192 (erinnern Sie sich noch an P=192 in Zeile 45? P wird nun hier benutzt).
FOR T = 1 TO 60:NEXT	Eine einfache Zeitverzögerungs-Schleife, die die Geschwindigkeit festlegt, mit der die Maus tanzt. (Probieren Sie eine höhere bzw. geringere Geschwindigkeit durch Erhöhung/Reduzierung der Zahl 60 aus.)

Zeile 85:

P=P+1	Nun erhöhen wir den Zeigerwert, indem wir den Originalwert P um 1 erhöhen.
IF P>194 THEN P=192	Wir wollen das Sprite nur auf drei Adreßbereiche zeigen lassen. 192 zeigt auf die Adressen 12288 bis 12350, 193 auf die Adressen 12352 bis 12414 und 194 auf die Adressen 12416 bis 12478. Diese Zeile weist den Computer an, P zurück auf 192 zu setzen, sobald P 195 wird. Auf diese Weise kann P nie wirklich 195 werden. P ist 192, 193, 194 und wird dann zurück auf 192 gesetzt. Der Zeiger zeigt nacheinander auf die drei Sprite-Formen in den drei 64-Byte-Gruppen der Adreßbereiche mit den Daten.

Zeile 90:

NEXT X	Nachdem das Sprite eine der drei durch die DATAs bestimmten Formen erhalten hat, kann es sich über den Bildschirm bewegen. Es überspringt jeweils drei X-Positionen (und bewegt sich nicht ruhig um jeweils eine Position weiter, was auch möglich ist). Hierdurch tanzt die Maus schneller über den Bildschirm. NEXT X schließt die Schleife FOR....X in Zeile 50 ab.

Zeile 95:

END	Beendet das Programm, wenn das Sprite sich aus dem Bildschirm hinausbewegt.

Zeilen 100–109:

DATA	Die Spriteformen werden nacheinander aus den DATA-Anweisungen gelesen. Zunächst werden die 63 Zahlen, die die Sprite-Form 1 enthalten, gelesen, danach die 63 Zahlen für Sprite-Form 2 und dann für Sprite-Form 3. Die Daten werden in die drei aufeinanderfolgenden Adreßbereiche gelesen. Nach dem Einlesen in diese Adressen braucht Sprite 0 lediglich noch auf die drei Speicherplätze zu zeigen. Das Sprite nimmt dann automatisch die entsprechende Form an. Da es auf diese Weise nacheinander entsprechend den Daten in den drei Sprite-Adressen unterschiedliche Formen annimmt, können wir einen Trickfilmeffekt erzeugen. Wenn Sie wissen wollen, wie diese Zahlen das einzelne Sprite beeinflussen, verändern Sie die ersten drei Zahlen in Zeile 100 zu 255, 255, 255. Wegen weiterer Einzelheiten schlagen Sie bitte im Abschnitt über die Definition der Sprite-Formen nach.

Zeile 200:

POKE S+4,129	Die auf 129 gesetzte Wellenformsteuerung schaltet den Geräuscheffekt ein.
POKE S+4,128	Die auf 128 gesetzte Wellenformsteuerung schaltet den Geräuscheffekt aus.
RETURN	Läßt das Programm zu Zeile 70 zurückspringen.

Zeile 300:

POKE S+11,129	Die auf 129 gesetzte Wellenformsteuerung schaltet den Geräuscheffekt ein.
POKE S+11,128	Die auf 128 gesetzte Wellenformsteuerung schaltet den Geräuscheffekt aus.
RETURN	Läßt das Programm zurück zum Ende von Zeile 75 springen.

Zusammenfassende Hinweise zur Sprite-Erstellung

In Bild 13 sehen Sie verschiedene Sprite-Speicher-Zeiger und Speicheradressen bei Verwendung des Kassettenpuffers. Dieser wird bei einem Reset des Computers allerdings gelöscht. Die Sprite-Definitionen gehen dann verloren.

Zudem bietet der Kassettenpuffer mit 200 Byte Größe nur Platz für 3 Sprites. Wollen Sie also animierte Sprites mit Sequenzen programmieren, bei denen zwischen vielen verschiedenen Sprite-Definitionen umgeschaltet wird, so empfiehlt es sich, einen anderen Bereich zu wählen.

Einschalten der Sprites

Durch POKE V+21,X (X=Zahl aus der Tabelle 15) können Sie jedes beliebige Sprite einschalten. Durch Einschalten von nur einem Sprite werden jedoch andere Sprites ausgeschaltet. Um zwei oder mehrere Sprites einzuschalten, müssen die Zahlen der betreffenden Sprites addiert werden. (Durch POKE V+21,6 werden z. B. die Sprites 1 und 2 eingeschaltet.) Nachstehend wird erklärt, wie Sie ein Sprite ein- und ausschalten können, ohne andere Sprites zu beeinträchtigen (besonders nützlich bei Trickfilmen).

Beispiel:

Um nur Sprite 0 auszuschalten, geben Sie ein: POKE V+21, PEEK V+21 AND (255–1). Ändern Sie die Zahl 1 in (255–1) in 1, 2, 4, 8, 16, 32, 64 oder 128 um (für die Sprites 0 bis 7). Um das Sprite wieder einzuschalten und nicht die bereits eingeschalteten übrigen Sprites zu beeinflussen,

Speichern im Adreßbereich (Zeiger setzen)	Sprite 0 2040,13	Sprite 1 2041,14	Sprite 2 2042,15	Wenn Sie 1 bis 3 Sprites benutzen, können Sie die Speicherplätze im Kassettenpuffer (832 bis 1023) benutzen. Bei mehr als 3 Sprites empfehlen wir jedoch, die Plätze 12288 bis 12798 (siehe Tabelle 15) zu benutzen
Sprite-Pixel-Adressen für Speicherblöcke 13–15	832 bis 894	896 bis 958	960 bis 1022	

Bild 13. Sprite-Zeiger und Adressen bei Verwendung des Kassetten-Puffers

Zum einfachen Konstruieren von Sprites								
	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7
Sprite einschalten	V+21,1	V+21,2	V+21,4	V+21,8	V+21,16	V+21,32	V+21,64	V+21,128
Speichern im Adreßbereich (Zeiger setzen)	2040, 192	2041, 193	2042, 194	2043, 195	2044, 196	2045, 197	2046, 198	2047, 199
Plätze für Sprite-Pixel (12288-12798)	12288 to 12350	12352 to 12414	12416 to 12478	12480 to 12542	12544 to 12606	12608 to 12670	12672 to 12734	12736 to 12798
Spritefarbe	V+39,C	V+40,C	V+41,C	V+42,C	V+43,C	V+44,C	V+45,C	V+46,C
Linke X-Position setzen (0–255)	V+0,X	V+2,X	V+4,X	V+6,X	V+8,X	V+10,X	V+12,X	V+14,X
Rechte X-Position setzen (0-255)	V+16,1 V+0,X	V+16,2 V+2,X	V+16,4 V+4,X	V+16,8 V+6,X	V+16,16 V+8,X	V+16,32 V+10,X	V+16,64 V+12,X	V+16,128 V+14,X
Y-Position setzen	V+1,Y	V+3,Y	V+5,Y	V+7,Y	V+9,Y	V+11,Y	V+13,Y	V+15,Y
Sprite horizontal (X) vergrößern	V+29,1	V+29,2	V+29,4	V+29,8	V+29,16	V+29,32	V+29,64	V+29,128
Sprite vertikal (Y) vergrößern	V+23,1	V+23,2	V+23,4	V+23,8	V+23,16	V+23,32	V+23,64	V+23,128
Setzen des Mehrfarbenmodus	V+28,1	V+28,2	V+28,4	V+28,8	V+28,16	V+28,32	V+28,64	V+28,128
Multicolorfarbe 1 (erste Farbe)	V+37,C	V+37,C	V+37,C	V+37,C	V+37,C	V+37,C	V+37,C	V+37,C
Multicolorfarbe 2 (zweite Farbe)	V+38,C	V+38,C	V+38,C	V+38,C	V+38,C	V+38,C	V+38,C	V+38,C
Setzen der Sprite-Prioritäten	Die Sprites mit der niedrigeren Zahl haben stets Vorrang vor den Sprites mit der höheren Zahl. So hat zum Beispiel Sprite 0 Vorrang vor allen anderen Sprites und Sprite 7 die letzte Priorität. Sprites mit niedrigeren Zahlen erscheinen daher stets vor Sprites mit höheren Zahlen.							
Kollision (zwischen Sprites)	V+30 IF PEEK(V+30) AND X=X THEN [action]							
Kollision (zwischen Sprites und Hintergrund)	V+31 IF PEEK(V+31) AND X=X THEN [action]							

Tabelle 15. Eine Übersicht zum einfachen Konstruieren von Sprites

geben Sie POKE V+21, PEEK(V+21) OR 1 ein und ändern Sie OR 1 in OR 2 (Sprite 2), OR 4 (Sprite 3) usw. um.

X-Positionswerte außerhalb von 255

X-Positionen reichen von 0 bis 255... und beginnen dann wieder bei 0. Um ein Sprite über die X-Position 255 hinaus bis an den rechten Bildschirmrand zu bewegen, ist zunächst die Anweisung POKE V+16,1 erforderlich. Dann wird ein neuer X-Wert von 0 bis 63 gePOKEt, der das Sprite in eine der X-Positionen auf der rechten Bildschirmseite setzt. Um zurück zu den Positionen 0 bis 255 zu gelangen, ist POKE V+16,0 und das POKEn eines X-Werts zwischen 0 und 255 erforderlich.

Y-Positionswerte

Die Skala der möglichen Y-Positionen von 0 bis 255. Hierbei liegen die Positionen 0 bis 49 über dem oberen Bildschirmrand, 50 bis 229 im sichtbaren Bereich und 230 bis 255 außerhalb des unteren Bildschirmrandes.

Spritefarben

Damit Sprite 0 seine Farbe ändert, geben Sie folgendes ein: POKE V+39,F (benutzen Sie für »F« die Farb-POKE-Eingabe aus Tabelle 15 sowie nachstehende Farb-Codes):

0 - schwarz	8 - orange
1 - weiß	9 - braun
2 - rot	10 - hellrot
3 - zyan	11 - dunkelgrau
4 - purpur	12 - mittelgrau
5 - grün	13 - hellgrün
6 - blau	14 - hellblau
7 - gelb	15 - hellbraun

Speicherplatz

Für jedes Sprite müssen Sie einen getrennten 64-Byte-Satz im Computerspeicher »reservieren«. Hiervon werden 63 Byte für die Sprite-Daten benutzt. Die nachstehend gezeigten Speicherbereiche entsprechen den Sprite-Zeigern in Tabelle 15. Jedes Sprite kann entsprechend Ihren Wünschen definiert werden. Wenn alle Sprites gleich aussehen sollen, müssen ihre Sprite-Zeiger auf die gleichen Sprite-Daten zeigen.

Verwendung der Sprite-Pointer

Diese Sprite-Pointer-Angaben sind nur als Empfehlungen zu verstehen. Sie können Sprite-Zeiger beliebig im RAM-Speicher setzen. Werden sie jedoch zu »niedrig« im Speicher gesetzt, kann ein langes Basic-Programm Ihre Sprite-Daten überschreiben oder umgekehrt. Um ein besonders langes Basic-Programm vor einer Überschreibung durch Sprite-Daten oder umgekehrt zu schützen, können die Sprites in einem höheren Speicherbereich abgelegt werden (z. B. POKE 2040,192 für Sprite 0 an den Adressen 12288 bis 12350... POKE 2041,193 an den Adressen 12352 bis 124,14 für Sprite 1 usw.). Durch geschickte Wahl der Speicheradressen, aus denen die Sprites ihre Daten empfangen, können Sie 64 verschiedene Sprites sowie ein ansehnliches Basic-Programm zusammen benutzen.

Definieren Sie hierfür verschiedene Sprite-Formen in Ihren DATA-Anweisungen und definieren Sie dann ein bestimmtes Sprite neu, indem Sie den Zeiger so ändern, daß für das betreffende Sprite verschiedene Speicherbereiche mit verschiedenen Sprite-Daten benutzt werden. Sehen Sie sich hierzu auch das Programm »Tanzmaus« (Listing 21) an. Sollen zwei oder mehrere Sprites die gleiche Form haben (Sie können immer noch Position und Farbe jedes Sprite ändern), benutzen Sie den gleichen Sprite-Zeiger und damit den gleichen Speicherbereich für die betreffenden Sprites (so können z. B. die Sprites 0 und 1 auf den gleichen Speicherplatz zeigen. Hierzu dient die Anweisung POKE 2040,192 und POKE 2041,192).

Prioritäten

Priorität bedeutet, daß ein Sprite vor oder hinter einem anderen Sprite auf dem Bildschirm gezeigt wird. Sprites mit höherer Priorität erscheinen stets vor (bzw. über) den Sprites mit niedriger Priorität. Hierbei haben Sprites mit niedrigerer Nummer stets den Vorrang vor solchen mit höherer

Nummer. Sprite 0 hat Priorität über alle anderen Sprites und Sprite 7 die niedrigste Priorität. Entsprechend hat Sprite 1 Vorrang vor den Sprites 2 bis 7. Befinden sich zwei Sprites an der gleichen Bildschirmposition, so erscheint das mit der höheren Priorität vor dem mit der niedrigeren. Das Sprite mit der niedrigeren Priorität ist entweder verdeckt oder »scheint durch«.

Arbeiten im Mehrfarbenmodus

Sie können mehrfarbige Sprites erstellen. Im Mehrfarbenmodus müssen Sie jedoch statt einzelner Punkte in Ihrem Spritebild stets Pixel-Paare benutzen (das heißt, jeder farbige »Punkt« oder »Block« im Sprite besteht aus mindestens zwei nebeneinanderliegenden Pixel). Es stehen vier Farben zur Auswahl: Spritefarbe (siehe Tabelle 15), Hilfsfarbe 1, Hilfsfarbe 2 und »Hintergrundfarbe« (die Hintergrundfarbe wird durch das Pixel-Paar 00 angewählt. In diesem Fall scheint der Hintergrund durch). Betrachten Sie einen horizontalen 8-Pixel-Satz in einem Spritemuster. Je nachdem, ob das linke, rechte oder beide Pixel ausgefüllt sind, wird die Farbe jedes Pixel-Paares bestimmt (siehe Bild 14).

HINTERGRUND	(Wenn beide Pixel leer (0) sind, scheint die Bildschirmfarbe durch.)
MEHRFARBIG 1	(Wenn das rechte Pixel in einem Pixel-Paar ausgefüllt ist, werden beide in der Hilfsfarbe 1 dargestellt.)
SPRITEFARBE	(Wenn das linke Pixel in einem Pixel-Paar ausgemalt ist, haben beide die Spritefarbe.)
MEHRFARBIG 2	(Wenn beide Pixel in einem Pixel-Paar ausgemalt sind, werden beide in der Hilfsfarbe 2 dargestellt.)

Bild 14. Pixel-Paare und ihre Farbquellen bei Multicolor-Sprites

Sehen Sie sich die horizontale 8-Pixel-Reihe in Bild 15 an. Nach dem oben Gesagten erscheint das erste Pixel-Paar in der Hintergrundfarbe, das zweite in der Hilfsfarbe 1, und für das dritte gilt die Sprite-Farbe. Das vierte Paar erhält die Hilfsfarbe 2. Die Farbe der einzelnen Pixel-Paare hängt also davon ab, welche Bits in dem Paar ausgemalt und welche leer sind. Wenn Sie festgelegt haben, welche Farben Sie für die einzelnen Pixel-Paare wünschen, müssen die Werte der ausgemalten Pixel im 8-Pixel-Satz addiert und danach diese Zahl in den geeigneten Speicherplatz gePOKEt werden. Ist z. B. die 8-Pixel-Reihe in Bild 15 die erste Reihe in einem Sprite, die bei Speicherplatz 832 beginnt, so lautet der Wert der ausgemalten Pixel 16+8+2+1=27 und es gilt die folgende Anweisung: POKE 832,27.

Kollision

Über folgende Zeile können Sie erkennen, ob Sprites miteinander kollidiert sind:

```
IF PEEK(V+30) AND X=X THEN (hier Aktion eingeben).
```

Es wird dabei festgestellt, ob das Sprite X–1 mit irgendeinem anderen Sprite zusammengestoßen ist. X ist 1 für Sprite 0, 2 für Sprite 1 und so fort bis 128 für Sprite 7.

Um festzustellen, ob das fragliche Sprite mit einem Zeichen des Bildschirmhintergrundes zusammengestoßen ist, kann man diese Zeile verwenden:

```
IF PEEK(V+31) AND X=X THEN (hier Aktion vorsehen).
```

Benutzen von Grafikzeichen in Datenanweisungen:

Das kleine Programm in Listing 22 ermöglicht Ihnen die Erstellung eines Sprite mit Leerstellen und ausgemalten Kreisen (<SHIFT Q>) in DATA-Anweisungen. Die in die Sprite-Datenregister gePOKEten Sprites und Zahlen werden am Bildschirm angezeigt.

```
10 PRINT CHR$(147):FOR I=0 TO 63:POKE 832+
   I,0:NEXT                                  <117>
20 GOSUB 60000                               <040>
999 END                                      <239>
60000 DATA ".........QQQQQQQ........"         <047>
60001 DATA ".......QQQQQQQQQQQ......"         <056>
60002 DATA "......QQQQQQQQQQQQQ....."         <177>
60003 DATA "......QQQQQ...QQQQQ....."         <247>
60004 DATA ".....QQQQQ.QQQ..QQQQ...."         <054>
60005 DATA ".....QQQQQ.QQQ.QQQQQ...."         <084>
60006 DATA ".....QQQQQ.QQQ..QQQQ...."         <056>
60007 DATA "......QQQQQ...QQQQQ....."         <251>
60008 DATA "......QQQQQQQQQQQQQ....."         <183>
60009 DATA "......QQQQQQQQQQQQQ....."         <184>
60010 DATA "......Q.QQQQQQQQQ.Q....."         <039>
60011 DATA ".......Q.QQQQQQQ.Q......"         <204>
60012 DATA ".......Q..QQQQQ..Q......"         <059>
60013 DATA "........Q..QQQ..Q......."         <168>
60014 DATA "........Q..QQQ..Q......."         <169>
60015 DATA ".........Q..Q..Q........"         <174>
60016 DATA ".........Q..Q..Q........"         <175>
60017 DATA "..........QQQQQ........."         <174>
60018 DATA "..........QQQQQ........."         <175>
60019 DATA "..........QQQQQ........."         <176>
60020 DATA "...........QQQ.........."         <057>
60100 V=53248:POKE V,200:POKE V+1,100:POKE
      V+21,1:POKE V+39,14:POKE 2040,13       <245>
60105 POKE V+23,1:POKE V+29,1                <186>
60110 FOR I=0 TO 20:READ A$:FOR K=0 TO 2:T
      =0:FOR J=0 TO 7:B=0                    <201>
60140 IF MID$(A$,J+K*8+1,1)="Q" THEN B=1     <086>
60150 T=T+B*2↑(7-J):NEXT:PRINT T;:POKE 832
      +I*3+K,T:NEXT:PRINT:NEXT               <251>
60200 RETURN                                 <057>
```

Listing 22. Ein einfacher Sprite-Editor

27
16+8+2+1

128	64	32	16	8	4	2	1

Sieht im Sprite wie folgt aus

Hintergrundfarbe	Hilfsfarbe 1	Spritefarbe	Hilfsfarbe 2

Bild 15. Ein Beispiel für die Farbzuordnungen in einem Multicolor-Sprite

Um Ihnen einen Überblick zu geben, finden Sie in Tabelle 16 alle Register des VIC-II-Chip und ihre Bedeutung aufgeführt.

Mit dem erworbenen Wissen über die verschiedenen Grafikmodi des C 64, deren Programmierung und Sprites besitzen Sie das Rüstzeug, um weiter in die Materie der Computergrafik einzudringen.

Auf den folgenden Seiten werden Sie sowohl in die praktischen als auch die theoretischen Grundlagen der Computergrafik auf dem C 64 eingeweiht. Auch Programmierung von bewegter dreidimensionaler Grafik wird dann kein Buch mit sieben Siegeln mehr sein.

Theorie und Praxis

Ausführlich und detailliert erfahren Sie nun, wie man Punkte setzt, Linien zeichnet und Kreise auf den Bildschirm zaubert, in Basic und Maschinensprache. Neben den erforder-

Register	Adresse	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	53248	X-Position des Sprite Nr. 0. Dazu muß Register 16 beachtet werden.							
1	53249	Y-Position des Sprite Nr. 0							
2	53250	X-Position des Sprite Nr. 1. Auch dazu, wie zu allen folgenden Sprites, muß Register 16 beachtet werden.							
3	53251	Y-Position des Sprite Nr. 1							
4	53252	X-Position des Sprite Nr. 2 s.o.							
5	53253	Y-Position des Sprite Nr. 2							
6	53254	X-Position des Sprite Nr. 3 s.o.							
7	53255	Y-Position des Sprite Nr. 3							
8	53256	X-Position des Sprite Nr. 4 s.o.							
9	53257	Y-Position des Sprite Nr. 4							
10	53258	X-Position des Sprite Nr. 5 s.o.							
11	53259	Y-Position des Sprite Nr. 5							
12	53260	X-Position des Sprite Nr. 6 s.o.							
13	53261	Y-Position des Sprite Nr. 6							
14	53262	X-Position des Sprite Nr. 7 s.o.							
15	53263	Y-Position des Sprite Nr. 7							
16	53264	Spr. 7 msb X-Pos.	Spr. 6 msb X-Pos.	Spr. 5 msb X-Pos.	Spr. 4 msb X-Pos.	Spr. 3 msb X-Pos.	Spr. 2 msb X-Pos.	Spr. 1 msb X-Pos.	Spr. 0 msb X-Pos.
17	53265	msb des Raster-registers (Reg.18)	Schaltbit für veränderten Hintergrund-farbmodus 1 = einge-schaltet	Schaltbit für Hochauflö-sungsmodus 1 = einge-schaltet	Schaltbit für Bildschirm »aus« 0 = normaler Bildschirm 1 = Bildschirm-farbe gleich Hintergrund-farbe	Schaltbit für Zeilenzahl 0 = 24 Zeilen 1 = 25 Zeilen	Wert der Zeilenverschiebung in Y-Richtung beim Smooth-Scrolling		
18	53266	Rasterregister. Dazu kommt das msb in Bit 7, Register 17							
19	53267	Lichtgriffel X-Position							
20	53268	Lichtgriffel Y-Position							
21	53269	Ein- und Ausschalten von Sprites. 0=Sprite aus. 1=Sprite an Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
22	53270			Reset-Bit muß 0 sein, damit VIC-II-Chip ar-beitet	Schaltbit für Mehrfarbmodus 1 = einge-schaltet	Schaltbit für Spaltenzahl 0=38 Spalten 1=40 Spalten	Wert der Spaltenverschiebung in X-Richtung beim Soft-Scrolling		
23	53271	Sprite-Vergrößerung in Y-Richtung. 0=normale Größe. 1=doppelte Größe. Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
24	53272	Startadresse des Bildschirmspeichers				- Startadresse des Speicherbereichs, in dem die Zeichen als Punktmatrizen abzurufen sind - Startadresse der Bit-Map			
25	53273	Interrupt-Flaggen-Register Interrupt				Lichtgriffel-In-terrupt-Flagge	Sprite/Sprite-Kollision	Sprite/Hinter-grund-Kollision	Raster-Inter-rupt-Flagge
26	53274	Interrupt-Masken-Register Interrupt				Lichtgriffel-In-terrupt-Maske	Sprite/Sprite-Kollision-Maske	Sprite/Hinter-grund-Kollision-Maske	Raster-Inter-rupt-Maske
27	53275	Sprite/Hintergrund-Prioritätenregister. 0=Sprite hat Priorität. 1=Hintergrund hat Priorität Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
28	53276	Sprite-Mehrfarbenmodus-Register. 0=Normalstellung. 1=Mehrfarbenmodus-Darstellung Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Spirte 0
29	53277	Sprite-Vergrößerung in X-Richtung. 0=normale Größe. 1=doppelte Größe Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
30	53278	Sprite/Sprite-Kollision. 0=keine Berührung. 1=Berührung Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
31	53279	Sprite/Hintergrund-Kollision. 0=keine Berührung. 1=Berührung Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0

Tabelle 16. Eine Übersicht über die Register des VIC-II-Chip (Fortsetzung auf der nächsten Seite)

Register	Adresse	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
32	53280	unbenutzt				Farbe des Bildschirmrahmens			
33	53281	unbenutzt				Hintergrundfarbe Nr. 0 (normale Hintergrundfarbe)			
34	53282	unbenutzt				Hintergrundfarbe Nr. 1			
35	53283	unbenutzt				Hintergrundfarbe Nr. 2			
36	53284	unbenutzt				Hintergrundfarbe Nr. 3			
37	53285	unbenutzt				Sprite-Mehrfarben-Register Nr. 0			
38	53286	unbenutzt				Sprite-Mehrfarben-Register Nr. 1			
39	53287	unbenutzt				Sprite 0, Farbe			
40	53288	unbenutzt				Sprite 1, Farbe			
41	53289	unbenutzt				Sprite 2, Farbe			
42	53290	unbenutzt				Sprite 3, Farbe			
43	53291	unbenutzt				Sprite 4, Farbe			
44	53292	unbenutzt				Sprite 5, Farbe			
45	53293	unbenutzt				Sprite 6, Farbe			
46	53294	unbenutzt				Sprite 7, Farbe			

Tabelle 16. Eine Übersicht über die Register des VIC-II-Chip (Schluß)

lichen Grundlagen lernen Sie auch noch, neue Basic-Befehle in den Interpreter einzubinden.

Bisher haben wir gesehen, wie man den C 64 in den verschiedenen Grafikmodi betreiben kann. Nun geht's in die Praxis! Sowohl in Basic als auch in Assembler werden wir nun gemeinsam eine kleine Bibliothek der wichtigsten grafischen Prozeduren aufbauen, die – besonders in Assembler – nach Möglichkeit auch noch optimiert ist, was die Geschwindigkeit der Ausführung angeht. Das Ergebnis unserer Arbeit sind dann einige neue Basic-Befehle, die für den abschließenden Teil des Kurses nützlich sein werden.

Grafik einschalten

Der erste Schritt, nämlich das Umschalten des Videocontrollers (VIC) in den Hochauflösungsmodus, erfordert zuerst ein wenig Planung. Wir haben ja in diesem Betriebszustand sowohl über den Bildschirm mit seinen 40 x 25 Zeichen, also 1000 Byte, als auch über die Bitmap mit ihren 320 x 200 Bit (also 8000 Byte) die volle Kontrolle. Insbesonders steht es uns frei, ihre Lage im Speicher so festzulegen, daß wir ein Optimum an Leistungsfähigkeit aus unserem Computer herausholen können. Sehr häufig findet man die Bitmap in Grafikprogrammen ab der Speicherstelle $2000 (also dezimal 8192). Die Lage des Bildschirms ist unverändert bei $0400 (dezimal 1024). Das ist zwar von der Umschaltung her recht bequem – wie wir gleich noch sehen werden –, hat aber einige Nachteile. Dadurch nämlich, daß der Textbildschirm im Hochauflösungsmodus als Farbspeicher benutzt wird, ist es nötig, alle Texte darauf zu löschen und gegebenenfalls nach dem Zurückschalten wieder neu zu schreiben. Ein weiteres Manko liegt in der Einschränkung des Speicherplatzes für Basic-Programme. Entweder beläßt man den Basic-Start bei 2048 und begrenzt den Speicher bis 8192 – dann hat man eben nur 6 KByte für Programme, Variable, Arrays und Strings zur Verfügung – oder man verlegt den Basic-Start nach $4000 (dezimal 16384). Dies verursacht immer etwas Unbehagen, weil damit schon allerlei Pannen passiert sind; zudem verschenkt man so den ganzen Speicherplatz unterhalb der Bitmap.

Kommt es vor allem darauf an, über möglichst viel freien Speicherplatz zu verfügen, dann ist das RAM unter dem Basic-ROM – also ab $A000 (dezimal 40960) für die Bitmap gut geeignet. Der Bildschirm muß dann im Bereich zwischen $8000 und $9000 liegen, denn zwischen $9000 und $A000 befinden sich die sogenannten Geisterbilder des Zeichen-ROM, die sich störend auf die Grafik auswirken. Diesen Weg habe ich in der Grafik-Erweiterung Hires-3 eingeschlagen. Einen Nachteil, der sich gerade in der Geschwindigkeitsoptimierung auswirkt, hat dieser Weg aber: Für jeden Lesezugriff in der Bitmap muß man das Basic-ROM abschalten und es danach sofort wieder einschalten. Das kostet unnötig viel Zeit und erfordert eine besonders sorgfältige Programmierung.

Wohin also mit dem Bildschirm und der Bitmap? Es gibt theoretisch acht mögliche Bereiche dafür im Computer. Der VIC-Chip greift auf die vorhandenen 64 KByte immer in 16-KByte-Schritten zu, die wir Abschnitte nennen wollen. Im untersten Abschnitt 0 (er reicht von Speicherstelle $0000 bis $4000, und das ist auch der Abschnitt, auf den der VIC-Chip im Einschaltzustand immer zugreift) käme nur der obere Teil – also $2000 bis $4000 – in Frage, denn andernfalls würde man die für die Computerfunktion lebensnotwendige Zeropage überschreiben. Dieser Abschnitt scheidet für uns – wegen seiner Nachteile – aus, wie wir vorhin festgestellt haben. Auch der Abschnitt 2 (von $8000 bis $C000) ist als nicht optimal erkannt worden, denn entweder muß man die Bitmap oder den Bildschirm unter das Basic-ROM legen und verliert dann kostbare Zeit durch das ständige Umschalten zwischen ROM und RAM. Ein ähnliches Problem stellt sich mit dem Abschnitt 3 (von $C000 bis $FFFF), wo man sich mit den Ein- und Ausgabeadressen und dem Betriebssystem-ROM herumzuschlagen hätte. Wie schon gesagt: Es ist möglich, hier Bitmaps einzurichten, aber es kostet Zeit! Es bleibt also der Abschnitt 1 (von $4000 bis $8000 oder dezimal 16384 bis 32768). Legen wir die Bitmap in die obere Hälfte dieses Abschnitts und direkt darunter den Bildschirm, dann steht ein relativ großer Speicherraum für Basic-Programme zur Verfügung (nämlich von $0800 bis $5C00, also von dezimal

Der Einstieg in die Praxis

2048 bis 23552) und so entstehen keine Zeitprobleme mit den ROM-Bausteinen. Zwar verschenken wir etwa 8 KByte an Speicherraum (von $8000 bis $A000), diese lassen sich aber durch Maschinenprogramme oder als Zwischenspeicher für Werte noch nutzen (unter Umständen kann auch die ganze Grafik-Befehlserweiterung dort plaziert werden). Angesichts der nunmehr etwa 21 KByte Speicherraum für

Basic-Programme läßt sich dieser Verlust leicht verschmerzen.

Nach all diesen Überlegungen soll nun die Programmpraxis folgen:

a) Der Zugriff des Videocontrollers auf die einzelnen Abschnitte wird in einem Register des CIA-2-Chip gesteuert. In 56576 ($DD00, das ist der Port A) bestimmen Bit 0 und 1 den VIC-Zugriff. Auf den Abschnitt 1 erfolgt der Zugriff, wenn Bit 1 auf 1 und Bit 0 auf 0 gesetzt wird.

b) Wo der Bildschirm zu finden ist, wird dem Computer über das Register 53272 des VIC-Chip mitgeteilt. Bit 4 bis 7 legen dann fest, um wie viele Byte der Bildschirmstart gegenüber dem Beginn des aktuellen Abschnitts verschoben ist. In unserem Fall müssen diese Bits auf den Wert 0111 gesetzt werden.

c) Auch dem Betriebssystem muß die neue Lage des Bildschirms mitgeteilt werden. In der Speicherstelle 648 erwartet es die Pagenummer des Bildschirmstarts. Diese Nummer ergibt sich durch Division der Bildschirmstartadresse durch 256. In unserem Fall ist dies die Zahl 92.

d) Weil es in jedem Abschnitt zwei Möglichkeiten für einen Bitmap-Start gibt, wird in Bit 3 der schon vorhin benutzten Speicherstelle 53272 festgelegt, welche wir uns davon ausgesucht haben. Daß wir die obere Hälfte für die Bitmap nutzen, teilen wir durch eine 1 in diesem Bit mit.

e) Nach all diesen Steuerschritten können wir nun den Bitmap-Modus des VIC-Chip einschalten. Das geschieht durch Setzen des Bit 5 im VIC-Register 53265.

f) Um ganz sicher zu gehen, daß ein eventuell vorhandenes Basic-Programm bzw. die Variablen, Arrays und Strings nicht die Bitmap oder den Bildschirm »zerfleddern«, begrenzen wir noch den Basic-Speicher. Das Ende soll beim Bildschirmstart liegen. Deshalb müssen wir das MSB (High-Byte) dieser Startadresse (das ist die Pagenummer) in die Speicherstellen 52 und 56 eintragen.

In Bild 16 sehen Sie diese Schrittfolge in Form eines Flußdiagramms dokumentiert. Dabei sind die beiden Einträge in der Speicherstelle 53272 zusammengefaßt worden.

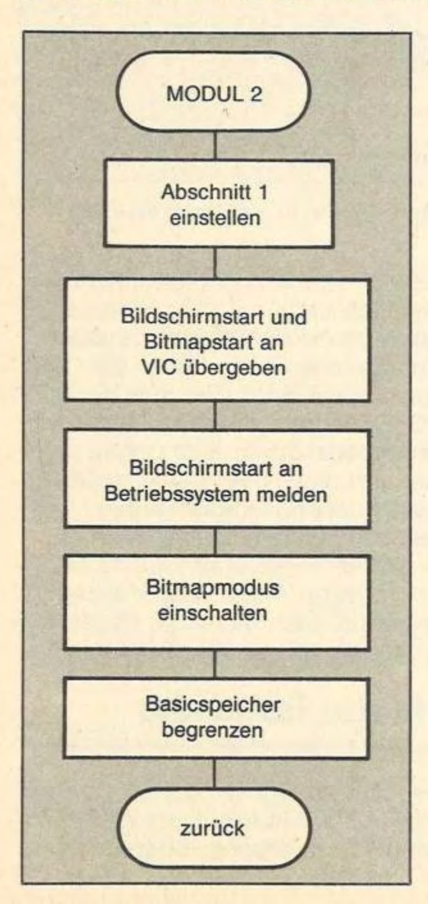


Bild 16. Flußdiagramm des Einschaltmoduls. In diesem Modul werden alle Speicherzellen initialisiert.

Diesen Teil unserer Aktivitäten habe ich Modul 2 genannt. In Basic sähe das Ganze dann etwa so aus:

```
2000 REM ----- MODUL 2: GRAFIK EINSCHALTEN
2010 POKE 56576,198:REM ABSCHNITT 1 EINSTELLEN
2020 POKE 53272,125:REM BILDSCHIRM- UND BITMAPSTART
     FESTLEGEN
2030 POKE 648,92    :REM BILDSCHIRMSTART AN
     BETRIEBSSYSTEM MELDEN
2040 POKE 52,92     :REM BASICSPEICHER BEGRENZEN
2050 POKE 56,92
2060 POKE 53265,PEEK(53265) OR 32:REM BITMAPMODUS
     EINSCHALTEN
2070 RETURN
```

Die Reihenfolge der einzelnen Schritte wurde noch etwas umgestellt, damit Sie den Vergleich zum Assembler-Programm besser erkennen. Dazu aber noch ein kleiner Kommentar: Die Grundbegriffe der 6502-Assembler-Sprache sollten Ihnen schon vertraut sein. Im Assembler-Programm – verwendet wird übrigens die Syntax des Assembler Hypra-Ass, – gehen diesem Modul 2 noch Modul 0 und Modul 1 voraus. Wir werden uns diese Programmteile später noch ansehen. Zur Erläuterung genügt es an dieser Stelle zu erwähnen, daß Modul 0 alle Zuweisungen von Werten oder Adressen an die im Programm verwendeten Label enthält, Modul 1 aber das Einbinden der neuen Befehle und ihrer Syntax in die Interpreterschleife vollzieht. Hier also nun das Modul 2 in 6502-Assembler:

```
2000 -     ;**** modul 2: grafik ein ****
2010 -     ;
2020 -     ;legt grafikbildschirm nach $5c00
2030 -     ;und bit-map nach $6000
2040 -     ;schaltet in bit-map-modus um
2050 -     ;und begrenzt basic bis $5c00
2060 -     ;
2070 -han       lda #$7d       ;ort bildschirm und
2080 -          sta vic24      ;bitmap festlegen
2090 -          lda #$c6       ;vic auf abschnitt 1
2100 -          sta cia2       ;richten
2110 -          lda #$5c       ;msb bildschirmstart
2120 -          sta hbase      ;an betriebssystem
2130 -          sta fretop     ;und speicher
2140 -          sta memsiz     ;begrenzen
2150 -          lda vic17      ;bitmapmodus
2160 -          ora #$20       ;einschalten
2170 -          sta vic17
2180 -          rts
```

Wie gesagt: Die Zuordnung aller vier verwendeten Namen (Label) geschieht in Modul 0. Sie erkennen sicherlich alle Elemente aus dem Basic-Modul wieder. Allerdings finden hier nur Zahlen des Hexadezimalsystems Verwendung. Nun wurde zwar die Grafik eingeschaltet, wenn Sie das aber einmal auf diese Weise tun, so werden Sie nur allerlei Müll auf dem Bildschirm entdecken. Wir brauchen also noch drei weitere Module:

Modul 3: Löschen der Bitmap
Modul 4: Farbgebung
Modul 5: Zurückschalten in den Textmodus

Erst danach können wir an die eigentlichen Zeichenroutinen, wie Punkt setzen oder löschen, Linie ziehen oder löschen, Rechtecke oder Kreise malen etc. gehen.

Löschen der Bitmap

In den Speicherzellen ab $6000 (dezimal 24256) befindet sich momentan noch ein zufälliges Bitmuster. Weil aber jedes gesetzte Bit in der Bitmap zu einem Punkt auf dem Bildschirm führt, wir aber eine saubere Zeichenfläche haben wollen, müssen wir in alle Bytes dieses Bereichs den Wert

0 schreiben. In Basic genügt dazu eine recht kurze Programmsequenz:

```
3000 REM ---- MODUL 3: BITMAP LOESCHEN
3010 BA = 24576 :REM DAS IST DIE BITMAPSTARTADRESSE
3020 FOR I = BA TO BA+7999 :REM DIE LOESCHSCHLEIFE
3030 :POKE I,0
3040 NEXT I
3050 RETURN
```

So kurz dieses Unterprogramm auch aussieht, so lange dauert seine Abarbeitung. Wenn Sie es aufrufen, können Sie in Ruhe eine Tasse Kaffee genießen, denn der POKE-Befehl ist ein recht langsamer Geselle. In Assembler geht das dagegen wie der Blitz. Bild 17 zeigt Ihnen ein Flußdiagramm dieses Moduls in der Assembler-Ausführung.

Hier schreiben wir zuerst die Bitmap-Startadresse in die Hilfs-Speicherstellen »help« und »help+1«, setzen dann den Akkumulator und das Y-Register auf 0 und schreiben ins X-Register die Anzahl zu löschender 256-Byte-Blöcke. Das sind 32, wobei Sie sicher bemerken werden, daß 32 mal 256 gleich 8192 ist. Wir beschreiben also 192 Byte zuviel. Das macht aber nichts, solange wir uns das merken und in dem Speicherbereich oberhalb von $8000 nicht in diesen ersten 192 Byte wichtige Werte aufbewahren. Schleifen, die eine exakte Durchlaufzahl bei angebrochenen Blöcken zulassen, sind vom Programmieraufwand – und daher auch von der Bearbeitungszeit im Programmlauf – wesentlich umfangreicher. Die Umsetzung dieses Moduls in 6502-Assembler entnehmen Sie bitte dem Programm Hires-4 (Listing 23 auf Seite 133).

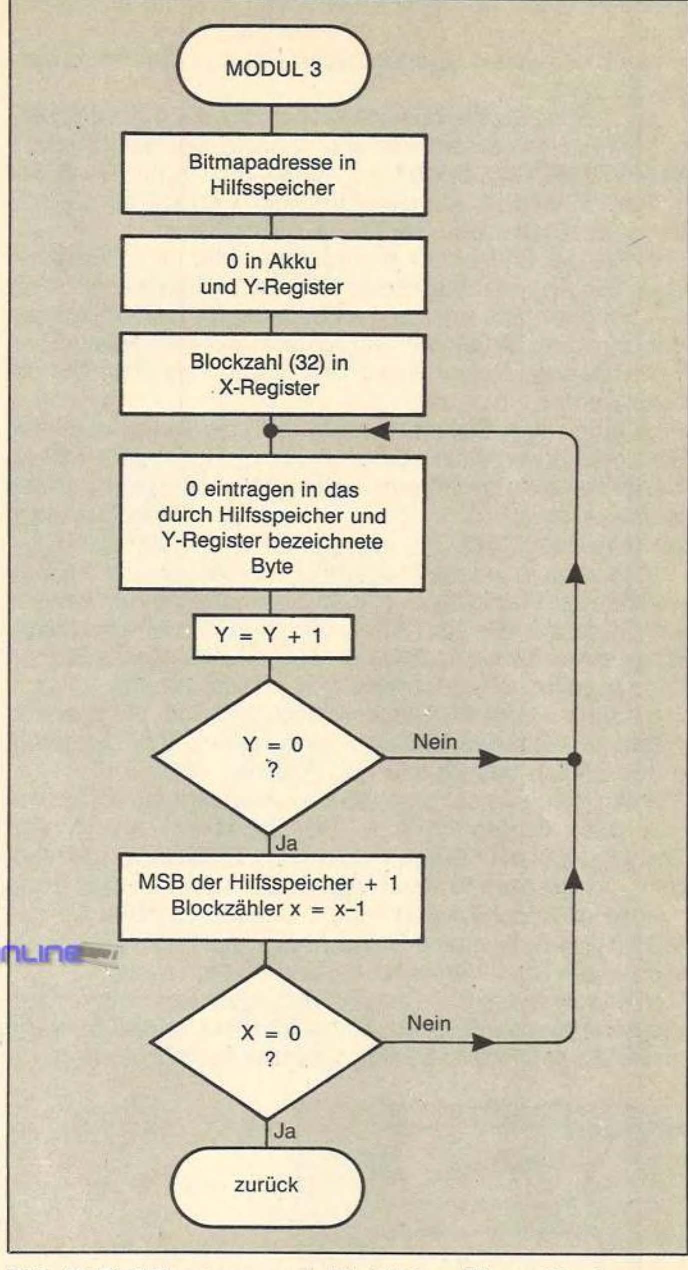


Bild 17. Flußdiagramm zum »Modul 3« – Bitmap löschen

Jetzt kommt Farbe in die Grafik

Das Modul 4 dient der Einrichtung der Farbkombinationen von Zeichenfarbe und Hintergrundfarbe. Wie Sie bereits aus dem Kurs wissen, dient der Bildschirmspeicher – den wir uns bei $5C00 eingerichtet haben – als Farbspeicher im Hochauflösungsmodus. Jede der 1000 Bildschirmspeicherzellen ist zuständig für ein Feld von 8 x 8 Bit der Bitmap. Der Inhalt eines solchen Bildschirmbytes steuert mit seinen unteren 4 Bit die Farbe des Hintergrundes und mit den oberen 4 Bit die Farbe der gesetzten Punkte. Die Farbcodes entsprechen den uns sonst schon bekannten im Textmodus.

Wenn man daher mit »ZF« die Farbe der Zeichnungen und mit »HF« die des Hintergrundes bezeichnet, dann kann man den Gesamtcode »F«, der in den Bildschirmspeicher zu schreiben ist, ausdrücken durch:

F = 16 * ZF + HF

Außer dieser Berechnung benötigt man für das Basic-Programm nur noch eine Schleife, um die Aufgabe des Moduls Nummer 4 zu erfüllen. Die Eingabe der gewünschten Farben muß allerdings schon vorher geschehen sein:

```
4000 REM ------ MODUL 4: FARBGEBUNG
4010 BS = 23552   :REM BILDSCHIRMSTARTADRESSE
4020 F = 16*ZF + HF :REM BERECHNUNG DES FARBCODE
4030 FOR I=BS TO BS+999 :REM FARBSCHLEIFE
4040 :POKE I,F
4050 NEXT I
4060 RETURN
```

Zu einer ganzen Tasse Kaffee reicht's hier nicht, ein paar Schlucke können Sie aber schon in aller Ruhe zu sich nehmen, bis dieses Programm abgearbeitet ist. Ein Flußdiagramm – hier wieder speziell für die Assembler-Version – zeigt Bild 18.

In den ersten Zeilen des Assembler-Programms wird mit Hilfe einer Interpreter-Routine namens »COMBYT« aus dem Basic-Text zuerst die Zeichenfarbe, dann die Hintergrundfarbe ins X-Register gelesen und von dort in Hilfs-Speicherzellen gelegt. Danach berechnet die Routine den Farbcode und schreibt schließlich in einer Doppelschleife diesen Code in den Bildschirmspeicher. Wieder ist die Anzahl der beschriebenen Speicherzellen höher als 1000 Byte (nämlich 4 x 256 Byte = 1024 Byte). Weil aber der Bildschirm an der Adresse 23552 beginnt, ist die letzte mit dem Farbcode beschriebene Adresse 24575, also genau unter der Bitmap-Startadresse, und dieses Vergehen richtet keinen Schaden an. Das Listing des Assembler-Moduls 4 finden Sie im Programmtext von Hires-4 (Listing 23).

Einen Nachteil hat dieses Verfahren, statt 1000 nun 1024 Byte mit dem Farbcode zu belegen: Falls Sie im Hochauflösungsbild auch mit Sprites arbeiten möchten, liegen die Spritezeiger noch in den oberen 8 Byte des 1024-Byte-Be-

Rückkehr in den Textmodus

reiches. Sie sollten daran denken, diese Spritezeiger erst nach dem Aufruf des Assembler-Moduls 4 mit Werten zu belegen, weil diese sonst mit dem Farbcode überschrieben werden.

Bild 18. Flußdiagramm zum »Modul 4« – Farbgebung

Von den Nicht-Zeichenbefehlen fehlt uns nun noch der, mit dem wir in den normalen Textmodus zurückkehren können. Das soll das Modul 5 leisten, in welchem die im Modul 2 veränderten Register wieder die Einschaltwerte erhalten und das Bit 5 in der VIC-Speicherstelle 53265 gelöscht wird. Als Ergebnis finden wir wieder den gewohnten Textbildschirm ab 1024 ($0400) mit seinem unveränderten Inhalt vor.

Nach dem Einschalten unseres Computers finden wir in der Speicherzelle 53272, mit der in Modul 2 der Bildschirm- und der Bitmap-Bereich festgelegt wurde, den Wert 21. Genau diesen schreiben wir nun dort wieder hinein. In der Speicherzelle 56576 haben wir den VIC-Chip auf den Abschnitt 1 gerichtet. Den Einschaltwert 199 schreiben wir nun dort hinein und teilen dem VIC damit wieder Abschnitt 0 zu. Die Speicherzelle 648 enthält den Wert 4. Dadurch wird dem Betriebssystem mitgeteilt, daß der Bildschirmspeicher bei der Adresse 1024 beginnt. Unverändert lassen wir die beiden Speicherstellen 52 und 56, mit deren Hilfe wir den Basic-Speicher verkleinert haben. Würden wir auch hier die Einschaltwerte eintragen, könnte das zum Verlust von Variablen oder Strings führen, wenn das aufrufende Progamm nach dem Aufruf des Moduls 5 weitergeführt werden soll. Durch Einführen neuer Strings oder Variablen hingegen könnten Bitmap oder Grafikbildschirm verändert werden. Schaltet man in diesem Fall im weiteren Programmlauf erneut in den Grafikmodus, fände man die Zeichnung zerstört.

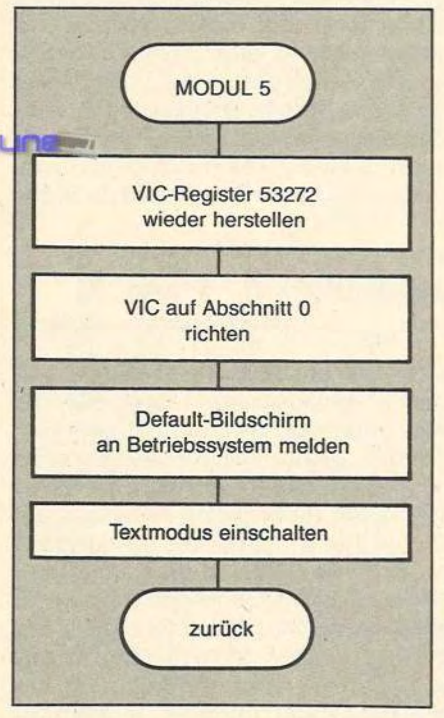


Bild 19. Flußdiagramm zum »Modul 5« – vom Grafikmodus in den Textmodus schalten

Sowohl die Basic- als auch die Assembler-Version des Moduls 5 wird erklärt durch das Flußdiagramm in Bild 19.

Die Basic-Version des Moduls lautet:

```
5000 REM -----MODUL 5: TEXTMODUS EIN
5010 POKE 53272,21 :REM ALTEN INHALT RESTAURIEREN
5020 POKE 56576,199:REM VIC WIEDER AUF ABSCHNITT 0
5030 POKE 648,4    :REM BILDSCHIRM AN
     BETRIEBSSYSTEM MELDEN
5040 POKE 53265,PEEK(53265) AND 223 :REM TEXTMODUS
     EINSCHALTEN
5050 RETURN
```

Die Assembler-Fassung von Modul 5 finden Sie im Listing Hires-4 (Listing 23). Sie sollten noch eine Maßnahme beachten: Vergessen Sie nicht, vor Beendigung Ihres Pro-

gramms den ursprünglichen Inhalt der Speicherstellen 52 und 56 wiederherzustellen - also den Wert 160 dort einzuschreiben -, das sollte man am Ende eines jeden Programms, das Routinen aus Hires-4 aufruft, tun, um die verbogenen Vektoren wiederherzustellen. Noch eine Bemerkung zur Speicherbegrenzung: Der Aufruf von Modul 2 muß erfolgen, bevor man den ersten String im Programm definiert. Es ist daher sinnvoll, Modul 2 sozusagen pro forma aufzurufen und dann mittels Modul 5 wieder in den Textmodus zu gehen. Geschieht das nämlich nicht, könnte es Probleme beim Wiederfinden von zuvor definierten Strings geben, die nach der Begrenzung im gesperrten Speicherbereich liegen.

Bevor wir nun zu den einzelnen Zeichenroutinen gelangen, sollen die Assembler-Module 1 und 0 noch erklärt werden. Außerdem werden wir damit automatisch auch noch dem Modul 11 begegnen, das in der Basic-Version nicht vorkommt.

An dieser Stelle soll nur das Prinzip erklärt werden, nach dem Modul 1 arbeitet, mehr würde den Rahmen dieses Kapitels sprengen.

Wenn unser Computer ein Basic-Programm bearbeitet, dann liest er es Byte für Byte durch. Jedes neu eingelesene Byte wird durch die sogenannte Interpreterschleife (das kommt von »interpretieren«) untersucht und das Ergebnis der Untersuchung führt dann beispielsweise dazu, daß in die PRINT-Routine des Basic verzweigt wird, wenn das Kennzeichen für den PRINT-Befehl gefunden wurde. Auf diese Schleife weist ein Zeiger, der »IGONE« genannt wird und der sich bei $308/9 befindet. Der Trick ist es nun, diesen Zeiger zu verbiegen und ihn auf eine eigene Interpreterroutine weisen zu lassen. Genau das ist der erste Schritt im Modul 1. Um also die ganze Befehlserweiterung zu aktivieren, muß zuerst immer dieser Teil - im Listing Hires-4 »init« genannt - angesteuert werden, was durch den Basic-Befehl SYS49152 geschieht (falls Sie Hires-4 bei $C000 belassen).

Die Assembler-Module 0, 1 und 11

Von nun an wird jedes Zeichen aus dem Basic-Text daraufhin überprüft, ob es zu einem unserer neuen Basic-Befehle gehört. Zu diesem Zweck haben wir als Modul 11 eine Reihe von Tabellen, unter denen sich auch alle neuen Befehlstexte befinden. Wird festgestellt, daß es sich nicht um einen von unseren selbstgebastelten Befehlen dreht, erfolgt ein Rücksprung in die normale Interpreterroutine. Andernfalls aber dient eine weitere Tabelle in Modul 11 dazu, die Startadresse der zum jeweiligen Befehl gehörenden Routine zu übergeben. Als Unterprogramm wird diese Routine dann bearbeitet und danach wieder in den normalen Interpreter zurückgesprungen. So haben wir gewissermaßen unsere eigene Interpreterschleife vor die normale Interpreterschleife geschaltet.

Der erste neue Basic-Befehl, der sich ebenfalls noch in Modul 1 befindet, lautet »AUS«. Er tut nichts weiter, als den Vektor »IGONE« wieder mit dem normalen Inhalt zu beschreiben, womit er unsere Befehlserweiterung abschaltet.

Sehen wir uns kurz noch das Assembler-Modul 0 an. Wie schon erwähnt, enthält es alle im Programm verwendeten Namenszuordnungen. Diese gruppieren sich in drei Abschnitte. Im ersten davon finden Sie Vektoren und Routinen des Interpreters und des Betriebssystems. Die verwendeten Namen sind in der Literatur allgemein gebräuchlich. Der zweite Teil des Moduls enthält ausschließlich Zeropage-Speicherstellen, die im Verlauf der verschiedenen Programmteile als Speicher für diverse Variable benutzt werden. Der dritte Abschnitt gibt noch eine genaue Übersicht über mehrfach verwendete Speicherstellen.

Das Tabellenmodul 11 hat folgenden Aufbau: Sie finden darin zuerst einige Konstanten im MFLPT-Format. Es handelt sich dabei um die Bildschirmgrenzen 199 und 319, wobei 199 auch als 1-Byte-Integer gespeichert wird. Es folgen einige Variable im MFLPT-Format, anschließend noch 2 Byte für das Modul 1 (zur Speicherung der Befehlsnummer und des Akkumulators). Ebenfalls mit dem Modul 1 arbeiten die Sprungtabelle (die angibt, wo die zu einem der neuen Basic-Befehle gehörende Routine steht) und die Befehlstabelle zusammen (in der alle neuen Basic-Befehle als ASCII-Texte enthalten sind). Beide Tabellen bieten noch Platz für Erweiterungen. Den Schluß des Moduls 11 bilden Rechentabellen, auf die wir bei der Erklärung der Zeichenmodule noch stoßen werden.

Punkte setzen oder löschen

Eine Routine zum Setzen von Punkten ist zweifellos die wichtigste im Rahmen eines Grafikprogramms. Gleichgültig, ob wir nun Kreise malen oder komplexe dreidimensionale Figuren: Letzten Endes setzt sich alles aus Bildschirmpunkten zusammen, die alle einzeln gesetzt werden müssen. Das macht uns der C 64 leider nicht gerade leicht. Die Organisation des Bildschirms im Hochauflösungsmodus haben Sie ja noch aus den Anfangsseiten dieses Kurses in Erinnerung, und von daher kennen Sie auch einen Weg, wie man aus der X- und der Y-Koordinate eines Punktes im Bildschirmsystem (also X von 0 bis 319 und Y von 0 bis 199) den Ort in der Bitmap berechnen kann, an dem ein Bit auf den Wert 1 zu setzen ist. Einen weiteren Weg (und zwar auf der Basis eines Artikels von Andreas Schömann) zeige ich Ihnen hier, er zeichnet sich besonders für das Assembler-Programm durch seine Schnelligkeit aus.

Nehmen wir einmal an, daß für die folgenden Überlegungen die Y-Koordinate eines Punktes gleich Null wäre und wir verändern nun nur die X-Koordinate. Was wir suchen, ist die Nummer des Bytes, in dem das Bit liegt, das wir verändern wollen. Geht man die oberste Bytereihe durch, dann steht zuerst links oben das Byte Nummer 0, rechts schließt sich Byte 8 an, dann Byte 16, 24, 32 usw. bis zum letzten Byte rechts oben mit der Nummer 312. Verfolgen wir nun einmal den Zusammenhang zwischen X-Koordinate und Byte-Nummer (wobei wir die letztere sowohl dezimal als auch als Binärzahl betrachten werden):

X-Wert	Byte-Nummer dezimal	binär
0 - 7	0	00000000 00000000
8 - 15	8	00000000 00001000
16 - 23	16	00000000 00010000
		...
312 - 319	312	00000001 00111000

Es zeigt sich, daß diese Nummern immer Vielfache von 8 darstellen, daß also Bit 0 bis 2 des X-Wertes (in Binärform beispielsweise 6 = 00000110) einfach auf Null gesetzt exakt zur Byte-Nummer führen. Genaugenommen betrifft das das LSB (das untere Byte) unseres X-Wertes, das MSB kann nur 0 (bei Werten kleiner als 256) oder 1 sein (bei Werten zwischen 256 und 319). Das Ausblenden von Bit 0 bis 2 kann man erreichen durch:

XL AND %1111 1000 (das ist dezimal 248)

Dabei ist XL das LSB der X-Koordinate. Sei XM das MSB, dann folgt für die Byte-Nummer:

256*XM + (XL AND 248)

Jetzt kommt die Y-Koordinate dazu. Nehmen wir für einen Moment einmal an, daß nun X so beschaffen sei, daß

wir uns in der ersten vertikalen Byte-Reihe befinden (also Werte zwischen 0 und 7 haben). Y kann sich zwischen 0 und 199 bewegen. Die Zählung der Byte-Nummern in der Senkrechten folgt – wie Sie wissen – einem 8-Byte-Rhythmus: 0, 1, 2, 3, 4, 5, 6, 7, 320, 321, 322, 323, 324, 325, 326, 327, ... Um festzustellen, in welcher dieser 8-Byte-Reihen wir uns befinden, genügt es, einfach Y durch 8 zu teilen und davon den Integerwert zu berücksichtigen. So führt uns ein Y-Wert von 8 zu INT(8/8) = 1. Das dazugehörige Byte liegt also in der ersten Reihe (wobei wir daran denken müssen, daß es auch noch eine nullte Reihe gibt!). In jeder dieser Reihen sind 320 Byte angeordnet. Um also die Nummer des ersten Byte einer solchen Reihe zu berechnen, ist die Nummer der Bytereihe noch mit 320 zu multiplizieren:

320 * INT(Y/8)

Für unseren Y-Wert 8 ergibt sich dann das Byte 320. Für Y=9 finden wir ebenfalls Byte 320 als erstes Byte der Reihe. Dazu muß aber nun noch ein Zählbetrag addiert werden, der sich immer zwischen 0 und 7 bewegt, der beispielsweise im Fall Y=9 genau 1 beträgt. Das kann erreicht werden durch den Ausdruck:

Y AND 7

7 lautet in binärer Form ja 0000 0111 und 9 ist binär 0000 1001. Die AND-Verknüpfung ergibt dann:

```
0000 1001    9
0000 0111    7
_____________ AND
0000 0001    1
```

Um also die vertikale Byte-Nummer herauszufinden, brauchen wir beide Ausdrücke nur noch zu addieren:

(YAND 7) + 320*INT(Y/8)

Damit haben wir alles, was wir zur Berechnung der Byte-Nummer benötigen. Insgesamt gilt nun für das zu verändernde Byte:

Byte-Nummer = 256*XM + (XL AND 248) + (Y AND 7) + 320*INT(Y/8)

Es kommt nun noch darauf an, festzustellen, welches Bit im so berechneten Byte zu beschreiben ist. Das können wir ähnlich feststellen, wie wir eben die vertikale Byte-Nummer berechnet haben, nämlich mit AND:

XL AND 7

Nehmen wir einmal an, unser X-Wert sei 8. Dann liefert 8 AND 7 den Wert 0. Das bedeutet aber, im entsprechenden Byte wird das 0. Bit – von links her gezählt – verändert! In der üblichen Zählweise – die ja von rechts arbeitet – handelt es sich um das Bit 7. Um die richtige Bit-Nummer zu erhalten, müssen wir daher schreiben:

Bit-Nummer = 7 - (XL AND 7)

Wie schalten wir nun ein solches Bit im berechneten Byte ein? In Basic genügt dazu die Zeile:

POKE Byte-Nummer, PEEK(Byte-Nummer) OR 2↑Bit-Nummer

Byte-Nummer hat hier schon die richtige Adresse, die sich durch Addition der berechneten Byte-Nummer und der Bitmap-Startadresse ergibt.

Spätestens hier trennen sich nun die Wege, die wir in Basic und in Assembler gehen. So wäre es in Basic unsinnig, die X-Koordinate in ein LSB und ein MSB zu teilen, weshalb hier die Berechnung der Byte-Nummer etwas anders durchgeführt wird:

Byte-Nummer = (X AND 504) + 40*(Y AND 248) + (Y AND 7)

Diese Formel ist aus der oben gezeigten ableitbar. Bei der Berechnung der Bit-Nummer ändert sich für das Basic-Programm lediglich, daß anstelle von XL hier X verwendet wird:

Bit-Nummer = 7 - (X AND 7)

Das komplette Modul 6 zum Setzen eines beliebigen Bildpunktes in Basic lautet damit:

```
6000 REM ----- MODUL 6: PUNKT SETZEN
6020 BY = (X AND 504) + 40*(Y AND 248) + (Y AND 7)
6030 BI = 7 - (X AND 7)
6050 POKE BY + BA, PEEK(BY + BA) OR (2↑BI)
6060 RETURN
```

Zwei Dinge gibt es nun noch, die im Assembler-Modul enthalten sind und hier in Basic ebenfalls sinnvoll erscheinen. Man sollte mittels Modul 6 auch einen Punkt löschen können. Es ist weiterhin wichtig, eine Kontrolle einzubauen, die verhindert, daß ein Punkt außerhalb des Bildschirmbereichs angesprochen wird.

Wenn das Modul 6 (beispielsweise aus einem Hauptprogramm) aufgerufen wird, müssen schon X und Y bekannt sein und mit übergeben werden. Wir sorgen nun nur noch dafür, daß auch eine Löschmarke L mit übergeben wird, die den Wert 0 hat, wenn ein Punkt zu setzen ist, aber den Wert 1, wenn er gelöscht werden soll. Das Modul prüft dann auf diese Löschmarke und setzt gegebenenfalls das berechnete Bit auf 0. Dies ist möglich mit der Kombination logischer Verknüpfungen AND NOT in der Zeile:

POKE BY + BA, PEEK(BY + BA) AND NOT (2↑BI)

Die Überprüfung darauf, ob sich ein Punkt innerhalb des Bildschirm-Koordinatensystems befindet, kann in einer Zeile geschehen:

IF X<0 OR X>319 OR Y<0 OR Y>199 THEN ...

Nach dem THEN erfolgt einfach ein Sprung zum Modul-

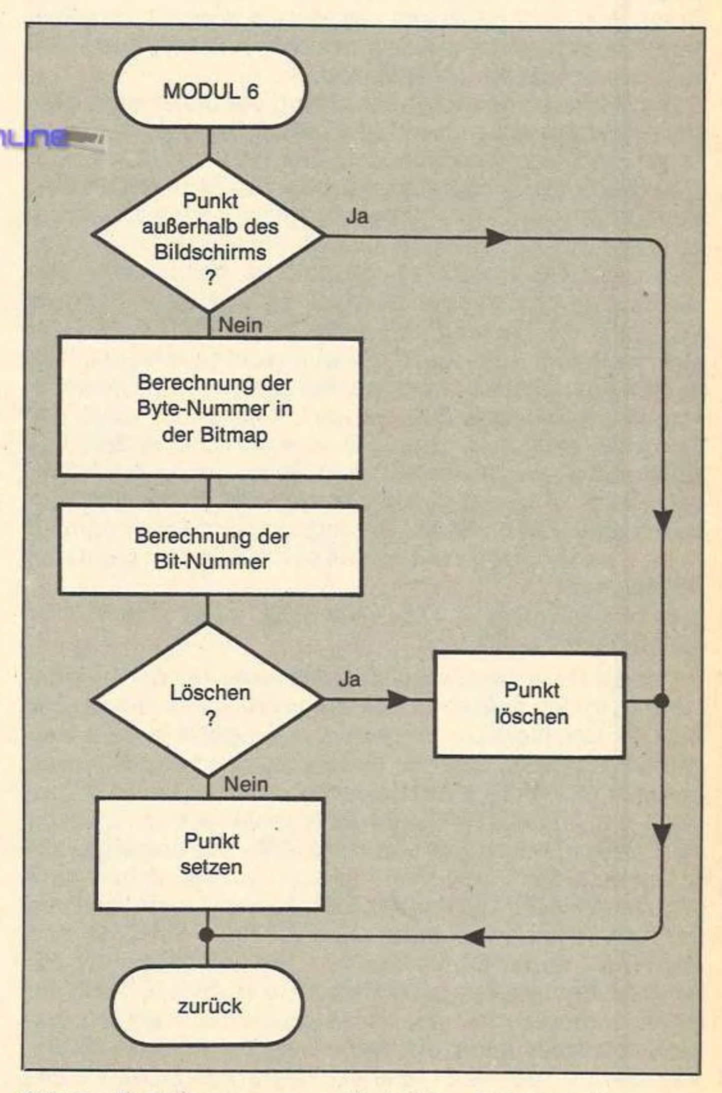


Bild 20. Flußdiagramm zum »Modul 6« – Setzen und Löschen von Punkten

ausgang (nämlich RETURN). Mit diesen Ergänzungen lautet das Basic-Modul 6 nun:

```
6000 REM ----- MODUL 6: PUNKT SETZEN / LOESCHEN
6010 IF X<0 OR X>319 OR Y<0 OR Y>199 THEN 6060
6020 BY = (X AND 504) + 40*(Y AND 248) + (Y AND 7)
6030 BI = 7 - (X AND 7)
6040 IF L=1 THEN POKE BY + BA, PEEK(BY + BA) AND
     NOT (2↑BI):GOTO            6060
6050 POKE BY + BA, PEEK(BY + BA) OR (2↑BI)
6060 RETURN
```

Zur Illustration unseres Vorgehens soll Ihnen das Flußdiagramm in Bild 20 dienen.

Sehen wir uns nun die Assembler-Fassung des Moduls an (siehe Listing 23 Hires-4). Vor allem in der Berechnung finden wir hier einige Unterschiede, die der Beschleunigung der Routine dienen. Gehen wir das Modul der Reihe nach durch.

Zunächst werden aus dem Basic-Text die Parameter des Punktes (X- und Y-Koordinate und Löschmarke) geholt. Dazu dient die Interpreterfunktion »GETNUM«, die einen 16-Bit-Wert nach »XKO«, »XKO+1« einliest und einen 8-Bit-Wert ins X-Register legt. Den letzteren merken wir uns in »YKO«. Die anschließend verwendete Interpreterroutine »COMBYT« holt ein Byte des Basic-Textes (nämlich unsere Löschmarke) ins X-Register, von wo aus wir es in die Speicherstelle »MODE« schieben. Es gibt da ein kleines Problem: Sind die Parameter X oder Y kleiner als 0 – was ja im Rahmen eines Programms schnell passieren kann –, dann erzeugen beide Interpreterroutinen eine Fehlermeldung. Damit das nicht geschieht, ist es also Aufgabe des Programmierers, vor dem Aufruf des Moduls sicherzustellen, daß X und Y größer als Null sind.

Die Prüfungen auf Überschreitung der maximalen Bildschirmwerte erledigt das Modul selbst. Dazu vergleicht es zuerst die Y-Koordinate in »YKO« mit 199. Ist »YKO« größer oder gleich dieser Zahl, dann wird verzweigt zu »NOPLOT«, also einfach zum aufrufenden Programm zurückgesprungen. Bei der X-Prüfung müssen 2 Byte getestet werden. Falls das MSB in »XKO+1« gleich 0 ist, kann einfach gezeichnet werden, ist es aber ungleich 0, wird es um 1 heruntergezählt. Spätestens nun sollte es gleich 0 sein. Falls aber nicht, war es zu groß und wird nicht gezeichnet. War es jetzt aber gleich 0, wird nun das LSB geprüft. Die maximale X-Koordinate ist 320, was im Hexadezimalsystem der Zahl $140 entspricht. Das LSB in »XKO« darf in dem Fall nicht größer oder gleich $40 sein. Ist es größer, wird nicht gezeichnet. Ansonsten befindet sich das Programm nun beim Label »OK«, und die Berechnung des Byte beginnt.

Zur Berechnung gehen wir von der anfangs vorgestellten Formel aus:

Byte = 256*Xm + (XL AND 248) + (Y AND 7) + 320*INT(Y/8) + BA

Der die Geschwindigkeit bestimmende Teil der Berechnung ist in den beiden letzten Summanden zu sehen. Der Trick ist nun, nicht per Programm diese ganzen Rechnungen auszuführen, sondern für jede der insgesamt 25 Reihen (zu je 8 x 40 Byte) schon vorher alle möglichen Ergebnisse von 320*INT(Y/8)+BA berechnet und in Tabellen »MALTAB« (für das LSB) und »MALTAB1« (für das MSB) abgelegt zu haben. Nun muß nur noch die Reihe bestimmt (eben durch INT(Y/8)) und dann der dazugehörige Wert aus den Tabellen gelesen werden. Die Tabellen »MALTAB« und »MALTAB1« finden Sie im Modul 11. Sie sind geordnet nach der Nummer der Reihe. So finden Sie in »MALTAB« für die Reihe 0 den Wert $00 und in »MALTAB1« den Wert $60. Insgesamt beginnt somit die Reihe 0 an der Adresse $6000, was der Startadresse unserer Bitmap entspricht. Im Berechnungsteil findet daher zunächst einmal die Division Y/8 statt, was durch die drei aufeinanderfolgenden LSR-Befehle geschieht. Automatisch ergibt sich dabei der Integerwert, denn Nachkomma-Bits werden einfach ignoriert. Das Ergebnis landet im Y-Register als Offset für das spätere Herauslesen aus den Tabellen. Der Y-Wert war zuvor noch im X-Register gespeichert und wird nun im Akku zu (Y AND 7) berechnet. Anschließend erfolgt die Addition des Tabellenwertes und noch die Addition des Gliedes (XL AND 248). In »ZWSP/ZWSP+1« liegt schließlich die Adresse des zu ändernden Byte fest.

Eine weitere Tabelle namens »HOCHTAB« enthält acht Zweierpotenzen, geordnet nach absteigender Größe. Wenn wir nun das zu ändernde Bit bestimmen – mit (XL AND 7) – können wir das Ergebnis dieser Berechnung im Y-Register direkt als Offset in die Tabelle »HOCHTAB« benutzen. Die Subtraktion 7-(HL AND 7) ist wegen dieser Reihenfolge der Tabelle nicht mehr nötig. Auf diese Weise ist nun auch der Wert bestimmt, der zur Veränderung des richtigen Bits führt.

Im weiteren Verlauf des Programms wird nun der Modus bestimmt (im Unterschied zum Basic-Modul wurde hier beim Löschen der Modus 0, beim Zeichnen aber 1 benutzt) und dann entsprechend verzweigt. Zum Setzen wird die ORA-Funktion verwendet, zum Löschen findet zuerst eine Invertierung des Bitwertes statt, und dann wird mit AND das Bit gelöscht.

Die gesamte Routine hat drei Einsprungstellen: »PUNKT 1« ist immer dann der Weg ins Modul, wenn die Parameter aus dem Basic-Text gelesen werden sollen. Das geschieht immer dann, wenn im Text der Befehl

PKT X,Y,Modus

auftritt, wobei X und Y größer als 0 sein müssen und Modus gleich 1 für Setzen und gleich 0 für Löschen festgelegt wurde. Über den Eingang »PUNKT 2« gelangt man ins Modul, wenn X und Y und Modus schon bekannt und in den Speicherstellen »XKO/XKO+1«, »YKO« und »MODE« abgelegt sind. Auch hier gelten die eben genannten Einschränkungen für die zugelassenen Werte aller drei Parameter. Die Prüfung auf Bereichsüberschreitungen erfolgt aber erst jetzt im Modul. Der dritte Einsprung – dann müssen alle Parameter auf ihre Zulässigkeit überprüft sein – kann bei »OK« erfolgen. Auch hier erwartet das Programm die Werte schon in den dafür vorgesehenen Speicherstellen.

Dieses im Assembler-Modul 6 verwendete Verfahren ist das schnellste zum Setzen oder Löschen von Punkten auf dem C64. Neben seinem Geschwindigkeitsvorteil hat es aber leider einen Nachteil, dem wir in den aufrufenden Programmen immer Rechnung tragen müssen: Es überprüft nicht, ob die Y-Koordinate kleiner als 0 ist. Man kann solch eine Werteüberprüfung in Assembler nämlich relativ einfach ausbauen zum sogenannten punkteweisen Clipping. Dazu wird bei jedem zu zeichnenden Punkt überprüft, ob seine Koordinatenwerte innerhalb eines zuvor definierten Bildschirmfensters liegen und das normale Bildschirmfenster (X von 0 bis 319, Y von 0 bis 199) wäre so nur ein Sonderfall. Ein Punkt wird nur gezeichnet, wenn er sich innerhalb des Fensters befindet. Auf negative X-Werte prüft Modul 6 übrigens gewissermaßen nebenher. Negative Integerzahlen sind im allgemeinen dadurch gekennzeichnet, daß ihr höchstwertiges Bit (also Bit 15 bei einer 2-Byte-Zahl) den Wert 1 aufweist. Das MSB unserer X-Koordinate würde dann beim Herunterzählen um 1 nicht 0 werden, und daher wird der Punkt auch nicht gezeichnet. Es ist dann nur noch die Aufgabe des Programmierers, dafür zu sorgen, daß in den aufrufenden Programmen negative Werte auch wirklich dieses Zahlenformat erhalten.

Linien zeichnen



Nachdem wir das Fundament eines Grafikprogramms geschaffen haben, gehen wir nun daran, die wichtigsten Zeichenroutinen zu programmieren. Zweifellos benutzt man am häufigsten einen Befehl zum Zeichnen von geraden Linien, und das soll für uns das Modul 7 erledigen.

Der einfachste Weg für den Basic-Programmierer wäre, mit der Funktionsvorschrift

Y = M * X + B

und dem DEF FN-Befehl eine X-Schleife durchlaufen zu lassen und die Punkte (X, FN(X)) mit Modul 6 zu zeichnen. Die Parameter M (Steigung) und B (Achsenabschnitt) wären entweder schon bekannt oder müßten im Programm aus den Koordinaten eines Start- und eines Endpunktes der Linie berechnet werden. Wenn wir dem Startpunkt den Namen P1(X1,Y1) und dem Endpunkt den Namen P2(X2,Y2) geben, dann könnte man das mit den folgenden Formeln tun:

M = (Y2-Y1)/(X2-X1)

B = -X1*(Y2-Y1)/(X2-X1) + Y1

Für die Mathematiker: Die beiden Gleichungen lassen sich aus der sogenannten 2-Punkte-Form der Geradengleichung herleiten. Leider verhält es sich so, daß dieser einfache Weg meist krumme Ergebnisse bringt. Die Bildschirm-

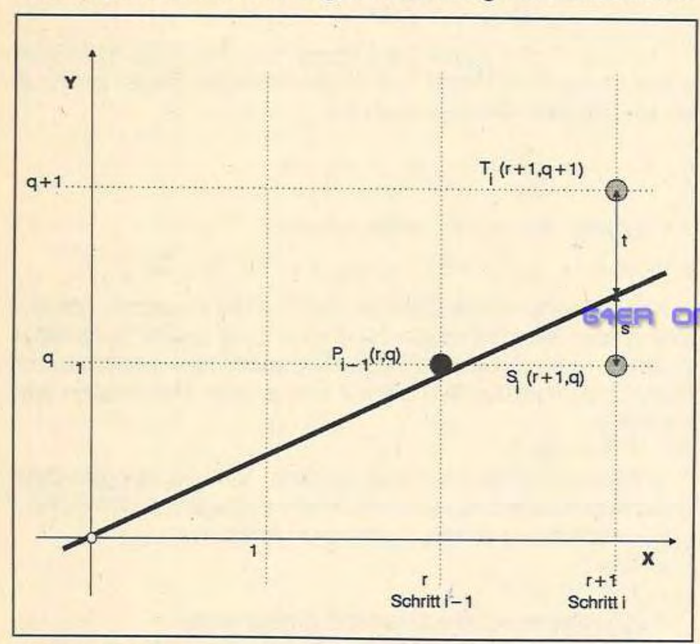


Bild 21. Der Bresenham-Linien-Algorithmus basiert auf der Ermittlung der Abstände t und s von der Ideallinie

koordinaten hingegen sind ganze Zahlen. Wenn sich durch die Berechnung nach dem obigen Algorithmus beispielsweise ein Y-Wert von 100,73 ergibt, kann nur bei Y=100 oder bei Y=101 ein Punkt gesetzt werden. Das gleiche gilt natürlich für die X-Koordinate. Das Modul 6 ist aufgrund seiner ganzen logischen Operationen nur auf Integerwerte eingerichtet. Es wird somit lediglich der Vorkommateil einer Zahl berücksichtigt, was auch dann der Fall wäre, wenn wir mit der INT(...)-Funktion gearbeitet hätten. Der Punkt in unserem Beispiel würde bei Y = 100 gesetzt, obwohl Y = 101 dem tatsächlichen Punkt näher läge. Das Ergebnis sind stark treppenartige Linien, die verhältnismäßig weit von der Ideallinie entfernt sind. Ein weiterer Hemmschuh ist die relativ lange Bearbeitungszeit, die dadurch hervorgerufen wird, daß bei jedem Schleifendurchlauf auf die Funktionsdefinition zurückgegriffen werden muß. Schließlich spielt es auch noch eine Rolle – besonders in der Assembler-Fassung –, daß nicht mit Integerzahlen gearbeitet werden kann. In den meisten Programmsprachen ist die Verarbeitung von Dezimalzahlen (also Kommazahlen) sehr viel aufwendiger als die von ganzen Zahlen.

Im Jahr 1965 hat Bresenham einen Linienalgorithmus veröffentlicht, der – in den zeitkritischen Programmteilen – ausschließlich ganze Zahlen verwendet und der außerdem den Treppeneffekt auf ein Minimum reduziert. Zudem ist er sehr schnell. Diesen Rechenweg werden wir verwenden für unser Modul 7. Es handelt sich dabei um einen sogenannten inkrementellen Algorithmus (vom lateinischen »increscere«, was »anwachsen, zunehmen« bedeutet). Das funktioniert so: Man geht von einem Startpunkt auf dem Bildschirm aus und geht in der X-Richtung um 1 weiter. Die Frage ist nun, wie muß man jetzt in der Y-Richtung weitergehen? Zum Y-Wert des Startpunktes ist ein gewisser kleiner Betrag – das Inkrement – zu addieren, damit man die Koordinaten des neuen Punktes erhält. Wenn wir das Inkrement »M« nennen, dann folgt auf diese Weise aus dem Startpunkt P1(X1,Y1) der nächste Punkt P2(X1+1,Y1+M). Die Größe dieses Inkrementes hängt ab von den typischen Parametern der Linie und kann aus diesen berechnet werden. Was wir bisher gezeigt haben, ist der einfache inkrementelle Algorithmus, der meistens noch Kommazahlen für »M« verwendet und von daher den Treppeneffekt aufweist. Bresenhams Weg untersucht nun für jeden neuen Punkt, welcher der möglichen Y-Werte den kleinsten Fehler aufweist. In Bild 21 sehen Sie die Verhältnisse noch etwas deutlicher dargestellt.

Optimierter Linienalgorithmus

Die Rasterpunkte des Bildschirms sind hier durch das Linienraster symbolisiert. Nur an den Kreuzungsstellen ist das Setzen von Punkten möglich. Die Linien haben in beiden Richtungen jeweils den Abstand 1 voneinander. Schräg durch das Bild verläuft das mathematisch korrekte Abbild einer Beispielgeraden. Ein Punkt dieser Geraden ist (tiefschwarz) schon gesetzt. Er weist die Koordinaten r und q auf. Das war der (i-1)-Punkt. Nun soll der i-Punkt gesetzt werden. Dazu gehen wir in X-Richtung um 1 weiter, so daß die X-Koordinate des i-Punktes nun r+1 beträgt. Es gibt (in diesem Beispiel) nun zwei mögliche Y-Positionen, nämlich q oder q+1. Die sich daraus ergebenden möglichen Punkte Si(r+1,q) und Ti(r+1,q+1) sind ebenfalls (grau) in Bild 21 zu sehen. Beide weichen von der idealen Linie ab, und zwar hat Si einen Fehler von s, Ti einen Fehler von t. Bresenhams Verdienst liegt darin, daß sein Algorithmus feststellt, welcher Fehler kleiner ist und dann zum Setzen des dazugehörenden Punktes führt.

Man kann sicher sagen, daß der Punkt Si zu setzen sei, wenn die Differenz s-t kleiner als 0 ist, und daß andernfalls Ti gesetzt werden muß. Wie kann man nun s und t auf einfache Weise ausdrücken? Gehen wir dazu noch einmal von der normalen Gleichung einer Geraden aus:

Y = M*X + B

Die Rechnung wird vereinfacht, indem man ein Koordinatensystem verwendet, in dem unsere Gerade genau durch den Ursprung verläuft (man nennt die dazu nötigen Operationen Transformationen, was uns aber momentan nicht beschäftigen soll). In diesem Fall ist der Achsenabschnitt B gleich 0, und die Gerade hat die vereinfachte Gleichung:

Y = M*X

M ist die Steigung der Geraden. Darunter versteht man den Tangens des Winkels, mit dem die Gerade ansteigt. Dieser Tangens kann berechnet werden, indem man an die Gerade ein rechtwinkliges Dreieck anlegt, so daß die Hypothenuse (also die lange Seite, die dem rechten Winkel gegenübersteht) durch unsere Gerade gebildet wird und die beiden Katheten (also die kurzen Seiten, die zusammen den rechten Winkel bilden) parallel zu den Koordinatenach-

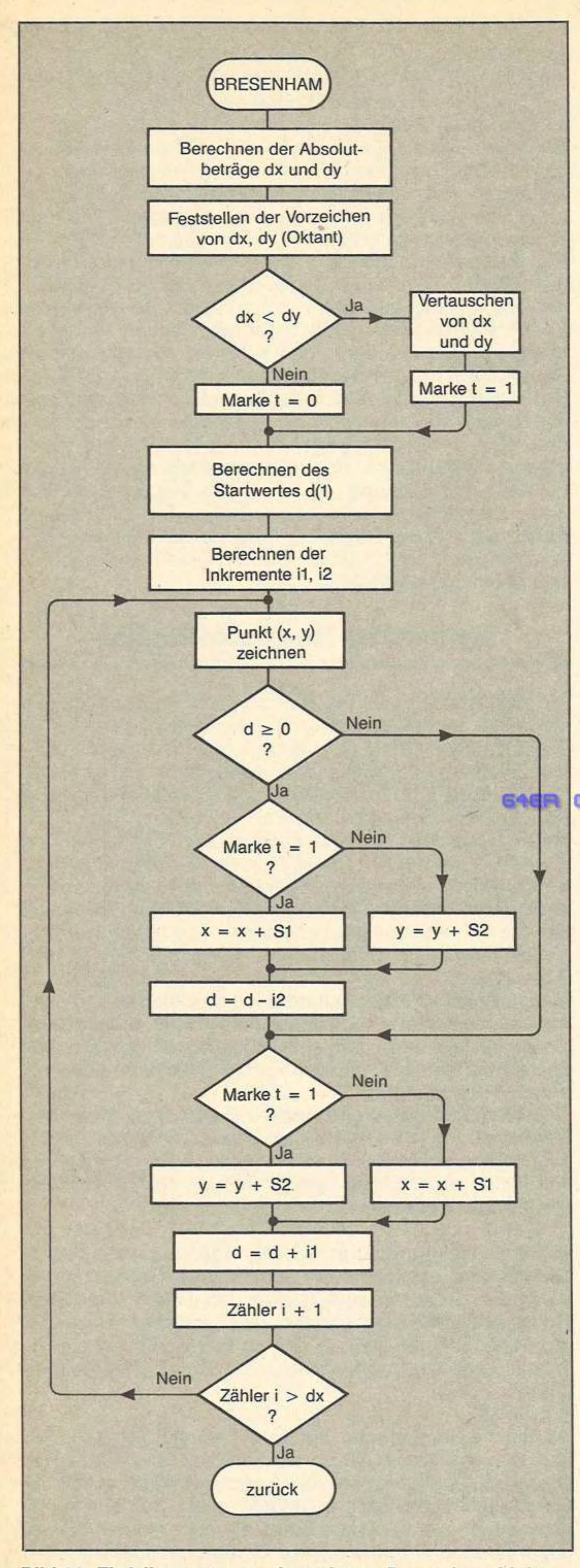


Bild 22. Flußdiagramm zum kompletten Bresenham-Linien-Algorithmus

sen verlaufen. Sind die Punkte, die die Hypothenuse einschließen, P1(X1,Y1) und P2(X2,Y2), dann kann man die Längen der Katheten dx und dy ausdrücken durch:

$dx = X2 - X1$

$dy = Y2 - Y1$

Der Tangens – also unsere gesuchte Steigung – ergibt sich dann zu:

$$M = \frac{dy}{dx}$$

Setzen wir das in die vereinfachte Geradengleichung ein, dann erhalten wir nun:

$$Y = \frac{dy}{dx} * X$$

Mit Hilfe von Bild 21 können wir nun s und t berechnen.
Für den Abstand s ergibt sich:

$$s = \frac{dy}{dx} * (r+1) - q$$

und für den Abstand t folgt:

$$t = q + 1 - \frac{dy}{dx} * (r+1)$$

Die uns interessierende Differenz s-t, die das Kriterium dafür ist, welcher Punkt nun zu zeichnen sei, ergibt sich aus diesen beiden Gleichungen zu:

$$s - t = 2 * \frac{dy}{dx} * (r+1) - 2 * q - 1$$

Das läßt sich leicht umformen in:

$dh * (s-t) = 2 * (r * dy - q * dx) + 2 * dy - dx$

Im uns gerade beschäftigenden Fall ist dx immer positiv. Wenn also s-t kleiner als Null wird (und daher Si gesetzt werden muß), dann wird auch die ganze linke Seite unserer Gleichung negativ. Wir geben dieser nun den neuen Namen d(i):

$d(i) = dx * (s-t)$

Außerdem führen wir nun anstelle von r und q die üblichen Koordinaten X und Y ein. Weil r und q im letzten Schritt (i-1) gewonnen wurden, setzen wir einfach:

$r = X(i-1)$

$q = Y(i-1)$

Aus unserer letzten Gleichung folgt nun:

(1) $d(i) = 2 * X(i-1) * dy - 2 * Y(i-1) * dx + 2 * dy - dx$

Betrachten wir nun den nächsten Schritt (i+1). Für diesen kann man analog dazu schreiben:

$d(i+1) = 2 * X(i) * dy - 2 * Y(i) * dx + 2 * dy - dx$

Jetzt ziehen wir d(i) von d(i+1) ab und erhalten für den Unterschied:

$d(i+1) - d(i) = 2 * dy * (X(i) - X(i-1)) - 2 * dx * (Y(i) - Y(i-1))$

Die Schrittweite in X-Richtung sollte aber immer 1 betragen, so daß gilt:

$X(i) - X(i-1) = 1$

Berücksichtigen wir das und lösen die Gleichung nach d(i+1) auf, dann erhalten wir:

$d(i+1) = d(i) + 2 * dy - 2 * dx * (Y(i) - Y(i-1))$

Jetzt können wir zwei Fälle unterscheiden:

a) $d(i) \geqq 0$, dann ist also $(s-t) \geqq 0$ und der Punkt Ti wird ausgewählt zum Zeichnen. Das bedeutet:

$Y(i) = Y(i-1) + 1$

und außerdem

(2) $d(i+1) = d(i) + 2 * (dy dx)$

b) $d(i) < 0$, dann ist $(s-t) < 0$ und der Punkt Si ist zu zeichnen. Hier ist dann:

$Y(i) = Y(i+1)$

und es gilt

(3) $d(i+1) = d(i) + 2 * dy$

Damit haben wir nun einen Weg gefunden, jedes d(i) aus dem vorhergehenden zu berechnen. Es fehlt uns nun noch ein Anfangswert d(1), der auf den Startpunkt zutrifft. Den können wir aus Gleichung (1) bestimmen, wenn wir für X(i-1) und Y(i-1) die Koordinaten des Startpunktes P0(0,0) einsetzen. Wir erhalten dann:

$d(1) = 2 * dy - dx$

Die gesamte Lösung läuft dann auf diese Weise:
1) Werte dx und dy berechnen
2) Anfangswert d(1) ermitteln
3) Feststellen, ob der aktuelle Wert d(i) größer oder kleiner als 0 ist
4) X um 1 erhöhen
5) Je nach Größe von d(i) Y beibehalten oder um 1 erhöhen
6) Je nach Größe von d(i) das nächste d(i+1) berechnen
7) Punkt zeichnen und weiter mit 3)

Wie Sie sehen, sind alle Werte als Integer-Zahlen verarbeitbar und im zeitkritischen Teil kommen nur Additionen vor. Damit ist das Optimum an Geschwindigkeit eines Linienalgorithmus erreicht.

Eine Einschränkung gibt es aber zu den bisher gemachten Überlegungen, die Ihnen sofort klar wird, wenn Sie in Gedanken einmal eine Gerade zeichnen lassen, die parallel zur Y-Achse verläuft: Es würde nichts gezeichnet werden, weil X nicht zu variieren ist! Der gezeigte Algorithmus arbeitet so nur für Linien im ersten Quadranten mit einem Steigungswinkel zwischen 0 und 45 Grad. Damit er für alle Eventualitäten brauchbar wird, müssen wir noch einige Ergänzungen anbringen – die aber am Prinzip nichts verändern. Die Ergänzungen haben die Aufgabe, die Steigung zu prüfen, gegebenenfalls statt X nun Y zu variieren und unter Umständen statt zu addieren auch zu subtrahieren. Den kompletten Algorithmus finden Sie als Flußdiagramm in Bild 22.

Kommen wir nun zur Basic-Version unseres Moduls 7. Wir gehen davon aus, daß die Koordinaten der beiden Linienbegrenzungspunkte (Startpunkte P1(X1,Y1) und Endpunkt P2(X2,Y2)) durch das Hauptprogramm schon bekannt sind. Außerdem soll vor dem Aufruf festgelegt werden, welche Größe (0 oder 1) die Löschmarke hat, denn wir können mit dem gleichen Modul auch Linien löschen (das erledigt dann Modul 6 für uns). Und so sieht unser Modul nun aus:

```
7000 REM ----- MODUL 7: LINIEN ZEICHNEN/LOESCHEN
7010 X% = X1:Y% = Y1 : REM ALLES IN INTEGERS
7020 DX% = ABS(X2-X1): REM BERECHNUNG DER
     DIFFERENZEN
7030 DY% = ABS(Y2-Y1)
7040 S1% = SGN(X2-X1): REM BESTIMMUNG DES OKTANTEN
7050 S2% = SGN(Y2-Y1)
7060 IF DX% < DY% THEN D%=DX%:DX%=DY%:DY%=D%:T%=
     1:GOTO 7080
7070 T%=0 : REM UMDREHEN WENN DX < DY UND
     MARKE SETZEN
7080 D% = 2*DY% - DX%: REM STARTWERT ERMITTELN D(1)
7090 I1% = 2*DY% : REM INKREMENTE BERECHNEN
7100 I2% = 2*DX%
7110 FOR I = 1 TO DX% : REM HAUPTSCHLEIFE
7120 :X=X%:Y=Y%:GOSUB 6000:REM PUNKT ZEICHNEN
7130 :IF D%(0 THEN GOTO 7170
7140 :IF T%=1 THEN X% = X% + S1%:GOTO 7160
7150 :Y% = Y% + S2%
7160 :D% = D% - I2%
7170 :IF T%=1 THEN Y% = Y% + S2%:GOTO 7190
7180 :Xe = X% + S1%
7190 :D% = D% + I1%
7200 NEXT I
7210 RETURN
```

Nach einem ähnlichen Prinzip ist auch die Assembler-Version des Modul 7 aufgebaut. Mit geringen Änderungen habe ich dazu ein hervorragendes Programm von Michael Bauer übernommen (siehe Listing Hires 4).

Wenn Sie unser Modul 7 benutzen, werden Sie feststellen, daß schräge Linien immer noch als Treppen erscheinen. Der Treppeneffekt hat nämlich zwei Ursachen: Eine Software-Ursache, die wir mittels Modul 7 minimieren konnten und eine Hardware-Ursache, die mit der relativ geringen Auflösung unseres Bildschirms zusammenhängt. Läßt man den gleichen Algorithmus beispielsweise in der höchsten Auflösung des Amiga (640 x 400 Punkte) laufen, ist kaum mehr eine Treppenzeichnung zu erkennen.

Kreis zeichnen

Zwar wird ein Befehl zum Zeichnen von Kreisen nicht mehr so häufig benutzt wie die bisher betrachteten, er zählt aber noch zu den wichtigeren Grafikbefehlen und soll deshalb im Modul 8 behandelt werden. Wieder gibt es einige herkömmliche und einfache mathematische Beziehungen, die oft zum Zeichnen von Kreisen herangezogen werden. Beispielsweise dient dazu:

$X = R * COS(W)$
$Y = R * SIN(W)$

W ist dabei ein Winkel, der – von 0 bis 360 Grad durchlaufen – die Punktekoordinaten eines Kreises mit dem Radius R liefert. Sie können sich sicher vorstellen, daß die Berechnungen des Sinus und des Cosinus sowie die Multiplikationen mit Realzahlen, langsam ablaufen und ähnliche Treppeneffekte hervorrufen, wie wir sie schon bei den Linienalgorithmen diskutiert haben. Aber auch hier gibt es erstaunlich schnelle Algorithmen, die noch dazu mit Integerwerten auskommen.

J. Michener hat (in Foley, Van Dam: »Fundamentals of interactive Computer Graphics«, Reading 1984, Seite 442) einen von Bresenham 1977 vorgestellten Kreis-Algorithmus verfeinert, den wir uns gleich genauer ansehen werden. Zuvor aber wollen wir noch einige Besonderheiten des Kreises untersuchen, die erheblich zur Beschleunigung der Rechnung beitragen können. Erfreulicherweise ist der Kreis eine höchst symmetrische Figur. Legt man ihn derart in ein Koordinatenssystem, daß sein Mittelpunkt mit dem Ursprung zusammenfällt, dann stellt man sehr schnell fest, daß es ausreicht, lediglich die Punkte eines Achtelkreises zu berechnen. Alle anderen lassen sich durch Symmetrien

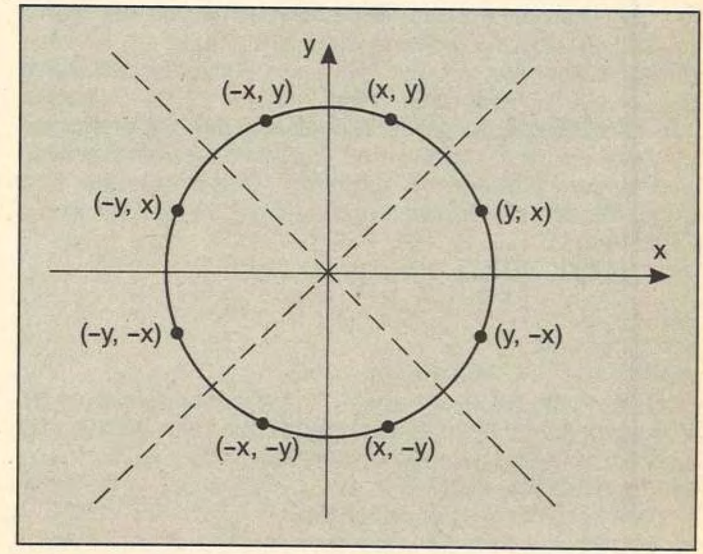


Bild 23. Zur Symmetrie des Kreises

daraus herleiten. In Bild 23 sehen Sie, daß aus einem berechneten Punkt sieben weitere Kreispunkte folgen.

Weil man aber prinzipiell jeden Kreis einer Transformation (also einer rein rechnerischen Verschiebung) unterwerfen kann, die ihn in die Ursprungslage bringt, braucht man immer nur ein Achtel der Kreispunkte zu berechnen. Nun also zu den Überlegungen von Bresenham und Michener. Bild 24 zeigt Ihnen die Verhältnisse in einem symbolisierten Bildschirmraster.

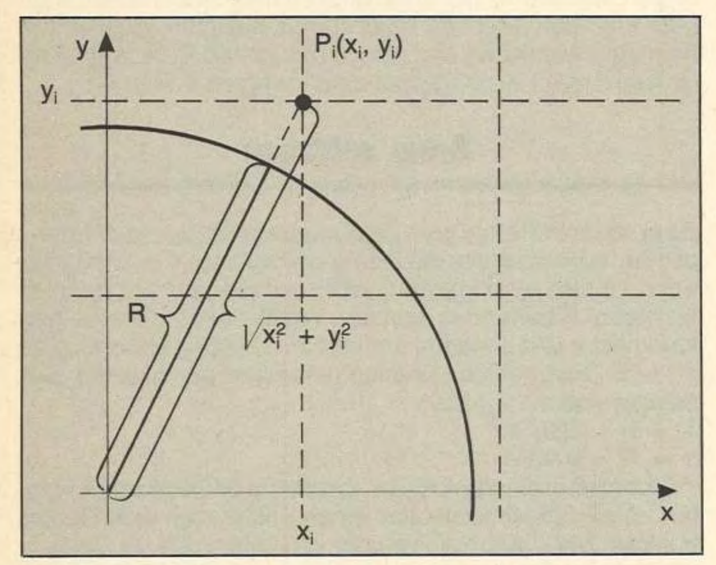


Bild 24. Der radiale Fehler beim Zeichnen eines Kreises

Sie sehen darin den idealen Kreis, seinen Radius R und einen Rasterpunkt Pi(Xi,Yi). Rechnen wir vom Mittelpunkt des Kreises aus, dann wäre die ideale Entfernung eines Kreispunktes R. Die wirkliche Entfernung aber des Punktes Pi vom Mittelpunkt ist SQR(Xi^2 + Yi^2). Gehen wir von der Gleichung des Kreises aus:

$X^2 + Y^2 = R^2$

Setzen wir darin einmal die wirkliche Entfernung ein und ein zweites Mal die Idealentfernung und ziehen beide Gleichungen voneinander ab, dann können wir eine Fehlergröße D(Pi) definieren, die auf einen Punkt Pi bezogen wird:

(1) $D(Pi) = (Xi^2 + Yi^2) - R^2$

Man könnte noch eine Reihe anderer Fehlergrößen festlegen – wir werden das später auch noch sehen –, Michener folgend nehmen wir hier den Radialfehler für die weiteren Betrachtungen. Sehen Sie sich dazu bitte Bild 25 an.

Wieder finden Sie hier das Koordinatenraster und einen Teil des idealen Kreises. Gerade eben wurde der Punkt P_{1-1} (X_{1-1}, Y_{1-1}) gesetzt. Wenn wir in X-Richtung um 1 weitergehen, stehen wir vor der Wahl, nun entweder den Punkt $S_1(X_{1-1} + 1, Y_{1-1})$ oder den Punkt $T_1(X_{1-1} + 1, Y_{1-1} - 1)$ zu setzen. Die Entscheidung darüber, welcher der beiden Punkte zu setzen ist, wird durch einen Vergleich der Fehlergrößen der beiden Alternativen getroffen. Diese ergeben sich durch Einsetzen der entsprechenden Koordinaten in die Gleichung (1):

(2) $D(S_1) = (X_{1-1} + 1)^2 + (Y_{1-1})^2 - R^2$

(3) $D(T_1) = (_{1-1} + 1)^2 + (Y_{1-1} - 1)^2 - R^2$

Wenn nun gilt:

$ABS(D(S_1)) >= ABS(D(T_1))$

dann ist T_1 der nächstgelegene Punkt, ansonsten ist es S_1. Wie zuvor schon beim Linienalgorithmus kann man wieder eine Entscheidungsgröße d_1 definieren:

(4) $d_1 = D(S_1) + D(T_1)$

Damit ergibt sich dann, daß wenn

$d_1 \geqq 0$ ist, der Punkt T_1 und wenn

$d_1 < 0$ ist, der Punkt S_1

gezeichnet werden muß. Nach einer ähnlichen Rechnung, wie wir sie für den Linienalgorithmus durchgeführt haben, erhält man schließlich für den Startwert der Entscheidungsgröße:

$d_1 = 3 - 2*R$

Hat man dann den Punkt S_1 gesetzt, weil d_1 kleiner als 0 war, dann ergibt sich für den nächsten Punkt die Entscheidungsgröße:

$d_{1+1} = d_1 + 4*X_{1-1} + 6$

Ist aber der Punkt T_1 gezeichnet worden, weil d_1 größer oder gleich 0 war, berechnet man die nächste Entscheidungsgröße nach:

$d_{1+1} = d_1 + 4*(X_{1-1} - Y_{1-1})$

Wie das nun in der Praxis aussieht, soll Ihnen das Flußdiagramm in Bild 26 zeigen.

Dieser Weg ist zwar schon praktikabel, wegen der vielen Rechnungen, die für D erforderlich sind, aber noch etwas langsam. Bresenham hat (in »Fundamental Algorithms for Computer Graphics«, NATO ASI Series f17, Berlin 1985, S.197) seine Arbeiten kritisch diskutiert und einige erhebliche Vereinfachungen veröffentlicht, die vor allem die Berechnung einer Fehlergröße betreffen. Am Prinzip ändert sich nichts. Der vereinfachte Algorithmus soll uns als Basis dienen für unser Modul 8.

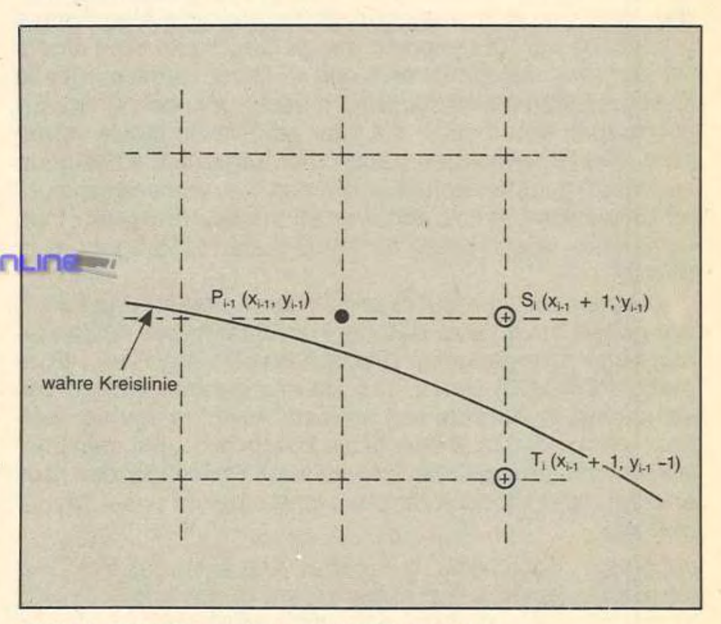


Bild 25. Zum Bresenham-Kreis-Algorithmus

Hier zunächst einmal die Basic-Version:

```
8000 REM ------- MODUL 8: KREIS ZEICHNEN/LOESCHEN
8010 X% = 0:Y% = R: REM STARTWERTE
8020 D% = R - 1 : REM ENTSCHEIDUNGSGROESSE
8030 IF D%<0 THEN Y% = Y% - 1:D% = D% + Y% + Y%
8040 GOSUB 8100 : REM ZEICHNEN VON 8 KREISPUNKTEN
8050 D% = D% - X% - X% - 1
8060 X% = X% + 1
8070 IF X%<=Y% THEN 8030
8080 RETURN
8100 REM ---- UP 8 KREISPUNKTE ZEICHNEN
8110 X = XM + X%:Y = YM + Y%:GOSUB 6000
8120 X = XM - X%:Y = YM + Y%:GOSUB 6000
8130 X = XM + X%:Y = YM - Y%:GOSUB 6000
8140 X = XM - X%:Y = YM - Y%:GOSUB 6000
8150 X = XM + Y%:Y = YM + X%:GOSUB 6000
8160 X = XM - Y%:Y = YM + X%:GOSUB 6000
8170 X = XM + Y%:Y = YM - X%:GOSUB 6000
8180 X = XM - Y%:Y = YM - X%:GOSUB 6000
8190 RETURN
```

Dem Modul müssen folgende Parameter durch das aufrufende Programm übergeben werden: Der Mittelpunkt des Kreises mit den Koordinaten XM und YM, der Radius R und die Löschmarke (1 = Löschen, 0 = Zeichnen). Das Unterprogramm enthält noch einige überflüssige Befehle (damit man zunächst einmal die Symmetriebeziehungen

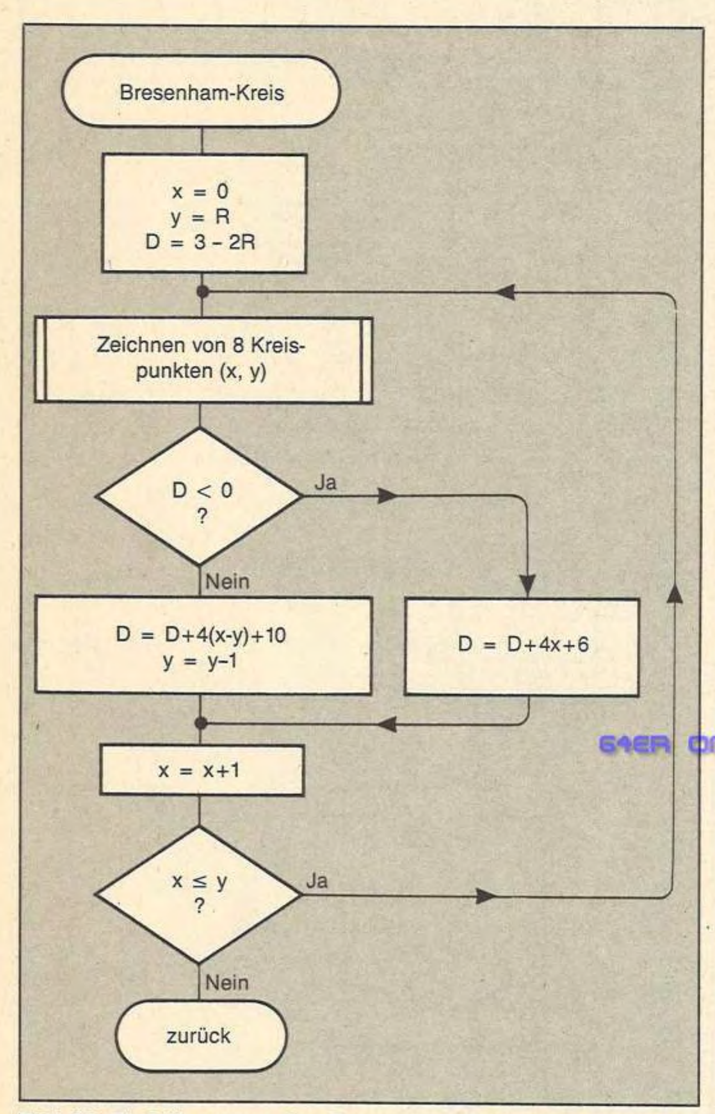


Bild 26. Flußdiagramm des Bresenham-Kreis-Algorithmus

erkennen kann), die durch entsprechende Anordnung noch eingespart werden können und damit das Modul schneller machen. Unter Umständen empfiehlt es sich auch, anstelle des Unterprogrammaufrufes in Zeile 8040 einfach die gesamten acht Zeichenbefehle direkt einzuarbeiten (in der verkürzten Form). Dann erspart man sich noch einmal überflüssige Programmzeit.

Die Assembler-Fassung des Moduls finden Sie im Listing 23 (Hires 4). Sie ist genauso aufgebaut wie das Basic-Modul. Hinzugekommen sind noch einige Zeilen, die die Parameter aus dem Basic-Text lesen und überprüfen. Noch eine Bemerkung zur Assembler-Fassung: Die verwendeten Speicherstellen sind hier fast vollständig für 16-Bit-Zahlen ausgelegt. Das rührt daher, daß diese Routine für spätere Erweiterungen offen sein soll. Solange sich alle Werte im normalen Bildschirmsystem befinden, gibt es mit der Kreisroutine keine Probleme. Sobald Sie aber Zahlen verwenden, die stark darüber hinausgehen – was aber ohnehin nicht sinnvoll ist – kann es zu unsinnigen Zeichnungen kommen.

Das Gesamtprogramm

Damit Sie alle bisher vorgestellten Grafikmöglichkeiten einmal ausprobieren können, sind zwei Listings angefügt. Hires 4.BAS (Listing 25) enthält alle Module und davor noch einen kleinen Testteil.

Wenn Sie das Programm starten, müssen Sie sich in Geduld fassen: Grafik in Basic ist auf dem C64 – wegen der vielen POKE-Befehle – langsam.

Etwa 25 Minuten lang können Sie Kaffee trinken oder zum Einkaufen gehen, während das Programm Punkte in Form einer Sinuskurve zeichnet, einen Teil davon wieder löscht, Linien als Radien eines Kreises zieht und zu guter Letzt einen großen Kreis auf den Bildschirm bringt. Danach sollten Sie sich einmal der Basic-Befehlserweiterung bedienen. Dazu laden Sie bitte das Programm Hires 4.OBJ (Listing 24) mit

```
LOAD"HIRES4.OBJ",8,1
```

Dann geben Sie ein NEW und CLR. Wenn Sie nun die Erweiterung durch

SYS 49152

aktivieren, stehen acht neue Basic-Befehle zur Verfügung:

AUS: Schaltet die Erweiterung wieder ab

HAN: Schaltet den Grafikmodus ein

LOE: Löscht die Bitmap

FAR,Zeichenfarbe,Hintergrundfarbe: Belegt den Grafikfarbspeicher mit den angegebenen Zeichen- und Hintergrundfarben

HOF: Schaltet wieder zurück in den Textmodus

PKT X,Y,L: Setzt oder löscht einen Punkt P(X,Y) auf dem Bildschirm. L ist die Löschmarke, die bei L=1 das Setzen und bei L=0 das Löschen bewirkt

LIN X1,Y1,X2,Y2,L: Zieht auf dem Bildschirm eine Linie von P1(X1,Y1) nach P2(X2,Y2). L ist wieder die Löschmarke

CIR,XM,YM,R,L: Zeichnet einen Kreis um den Mittelpunkt M(XM,YM) mit dem Radius R. Auch hier ist L die Löschmarke

Das gleiche wie Hires 4.BAS leistet auch das Programm Hires 4.ASS (Listing 26), nur eben mit der Basic-Erweiterung.

Nach dem Start müssen Sie sich aber etwas beeilen, schnell einen Schluck Kaffee hinunterzustürzen: Nach knapp 25 Sekunden ist das Programm beendet, und das liegt auch noch hauptsächlich an den Warte- und Zeichenschleifen in Basic, die unsere neuen Befehle ansteuern.

Zum Abschluß dieses Kapitels sei Ihnen noch gesagt, daß wir in der Befehls-Erweiterung lediglich ein Grundgerüst geschaffen haben.

Nun bleibt es Ihnen überlassen, weitere Anwendungen oder Besonderheiten in Basic damit zu programmieren.

So können Sie nun das eine oder andere Programm ein wenig umformulieren. Wenn Sie genau hinsehen, werden Sie feststellen, daß in den Tabellen des Moduls 11 (Assembler-Fassung) noch allerlei Erweiterungen vorgesehen sind.



```
ready.

10   -;
20   -;********************************
30   -;*                              *
40   -;*     Grafik in Assembler      *
50   -;*    Programm in  9 Modulen    *
60   -;* Grafik-Basic-Erweiterung C64 *
70   -;*           HIRES-4            *
80   -;*                              *
90   -;*  Heimo Ponnath Hamburg 1986  *
100  -;*                              *
110  -;********************************
120  -;
130  -;*** Modul Ø: Zuweisungen ***
140  -;
```

Listing 23. Quelltext im Hypra-Ass-Format zu Hires 4

```
150  -          .ba $cØØØ
160  -;
170  -;--- feste Zuordnungen ---
180  -;
190  -          .eq xko=$14    ;2-Byte X-Koordinate
200  -          .eq fretop=$34 ;Stringstartadresse MSB
210  -          .eq memsiz=$38 ;Hoechste Basicadresse MSB
220  -          .eq chrget=$73 ;CHRGET-Routine
230  -          .eq chrgot=$79 ;CHRGOT-Routine
240  -          .eq txtptr=$7a ;CHRGET-Zeiger
250  -          .eq hbase=$Ø288;MSB Bildschirmstart an Betriebssystem
260  -          .eq igone=$Ø3Ø8;Vektor zum Routinenaufruf
270  -          .eq bild=$5cØØ ;bildschirmstartadresse
280  -          .eq bitmap=$6ØØØ;Startadresse Bitmap
290  -          .eq gone1=$a7e4;Alter Inhalt von igone
300  -          .eq intend=$a7e7;Ende der Interpreterschleife
310  -          .eq linget=$a96b;holt 2-Byte-Integer nach $14/15
320  -          .eq frmnum=$ad8a;holt beliebigen numerischen Ausdruck
330  -          .eq chkcom=$aefd;auf Komma pruefen
340  -          .eq facinx=$b1aa;FAC -> Integer in A/Y
350  -          .eq getadr=$b7f7;FAC -> Integer in Y/A und $14/15
360  -          .eq getnum=$b7eb;holt Adresse nach $14/15 und Byte ins X-Reg.
370  -          .eq fsub=$b85Ø ;FAC = (A/Y) - FAC
380  -          .eq fmult=$ba28;FAC = (A/Y)*FAC
390  -          .eq fdiv=$bbØf ;FAC = (A/Y) / FAC
400  -          .eq movfm=$bba2;laedt FAC aus Speicher
410  -          .eq movmf=$bbd4;laedt Speicher aus FAC
420  -          .eq vic17=$dØ11;Bit 5 schaltet Bit-Map-Modus ein/aus
430  -          .eq vic24=$dØ18;Steuert Bildschirmort und Bit-Map-Start
440  -          .eq cia2=$ddØØ ;CIA2 PortA:Steuert VIC-Abschnitt
450  -          .eq combyt=$e2ØØ;Prueft auf Komma und holt 1 Byte in X
460  -;
470  -;--- Variable -----
480  -;
490  -          .eq mode=$Ø2   ;Modus Ø=loeschen, 1=zeichnen
500  -          .eq yko=$Ø9    ;y-Koordinate
510  -          .eq yk1=$Øa    ;Koordinatenwert y1
520  -          .eq dy=$11     ;Abstand y1-yØ
530  -          .eq iy=$19     ;Inkrementwert
540  -          .eq ay=$1a     ;   -"-
550  -          .eq ax=$1b     ;   -"-
560  -          .eq ix=$1c     ;   -"-
570  -          .eq yplusy=$57 ;Kreispunktspiegelwert
580  -          .eq yminy=$59  ;   -"-
590  -          .eq yplusx=$5b ;   -"-
600  -          .eq yminx=$5d  ;   -"-
610  -          .eq help=$fb   ;2-Byte Hilfswert
620  -          .eq zwsp=help+2;   -"-
630  -          .eq ct=$93     ;2-Byte-Zaehler
640  -          .eq rad=$75    ;kreisradius
650  -          .eq xplusx=$5f ;Kreispunktspiegelwert
660  -          .eq xminx=$9b  ;   -"-
670  -          .eq xplusy=$9e ;   -"-
680  -          .eq xminy=$a3  ;   -"-
690  -          .eq xk1=$8b    ;Koordinate x1
700  -          .eq ym=$8d     ;Kreismittelpunkt Y
710  -;
720  -;--- Doppelt genutzte Variable ---
730  -;
740  -          .eq of=xk1     ;Approximationswert
750  -          .eq dx=help    ;Abstand x1-xØ
760  -          .eq xm=xk1     ;Kreismittelpunkt X
770  -;
780  -;*** modul 1: Befehlserweiterung ***
790  -;
800  -;Erweitert die Interpreterschleife
810  -;um einen Teil, der die neuen
820  -;Basicbefehle prueft und zur
830  -;Ausfuehrung bringt. Ab INIT wird
840  -;diese neue Schleife eingerichtet,
850  -;ab AUS kann sie abgeschaltet werden.
860  -;Ab START liegt die eigentliche
870  -;Interpretererweiterung.
880  -;
890  -;
900  -;------ Initialisierung -------
910  -;
920  -init     lda #<(start)  ;lsb eigene Routine
930  -          sta igone      ;in vektor schreiben
940  -          lda #>(start)  ;msb
950  -          sta igone+1
960  -          rts
970  -;
980  -;------ Abschalten ------------
990  -;
1000 -aus      lda #<(gone1)  ;vektor auf
1010 -          sta igone      ;Normalwert
1020 -          lda #>(gone1)  ;zurueckstellen
1030 -          sta igone+1
1040 -          rts
1050 -;
1060 -;-Erweiterte Interpreterschleife-
1070 -;
1080 -start    jsr chrget     ;Zeichen holen
1090 -          cmp #$6Ø       ;Buchstabe?
1100 -          bcs ende       ;Basic-Code
1110 -          cmp #$41       ;Buchstabe A ?
1120 -          bcc ende       ;Sonderzeichen
1130 -          sta akku       ;Akku sichern
1140 -          ldx #$ØØ
1150 -          stx befnr      ;Befehlsnr. auf Ø
1160 -int1     ldy #$ØØ
1170 -          inc befnr      ;Befehlsnr. + 1
1180 -          lda beftab,x   ;Zeichen aus Befehlstabelle
1190 -          bne int2       ;kein Trennzeichen
1200 -;
1210 -          lda akku       ;Zurueck ins
1220 -ende     jsr chrgot     ;normale Basic
1230 -          jmp intend     ;springen
1240 -;
1250 -;--- Adresse suchen -----------
1260 -;
1270 -int2     cmp (txtptr),y ;Vergleich mit Basictext
1280 -          bne rest       ;ungleich
1290 -          iny            ;Basictextindex+1
1300 -          inx            ;Befehltab.-Index+1
1310 -          lda beftab,x   ;naechstes Zeichen
1320 -          bne int2       ;pruefen
1330 -          clc
1340 -          tya            ;Befehlsindex um
1350 -          adc txtptr     ;Befehlslaenge
1360 -          sta txtptr     ;erhoehen
1370 -          bcc lab1       ;Uebertrag?
1380 -          inc txtptr+1   ;msb erhoehen
1390 -lab1     lda befnr      ;Befehlsnr.
1400 -          asl            ;verdoppeln
1410 -          tax            ;und als Index in
1420 -          lda sprtab,x   ;Sprungtabelle
1430 -          sta sprung+1   ;lsb Sprung
1440 -          lda sprtab1,x  ;msb lesen
1450 -          sta sprung+2   ;msb
1460 -;
1470 -;-- Selbstmodifizierender Teil --
1480 -;
1490 -sprung   jsr $ffff      ;Dummy
1500 -;
1510 -;-- Zurueck zum Interpreter -----
1520 -;
1530 -          jmp ende
1540 -;
1550 -;-restl. Befehlstext ueberlesen--
1560 -;
1570 -rest     inx
1580 -          lda beftab,x
1590 -          bne rest       ;bis Trennzeichen
1600 -          inx
1610 -          jmp int1       ;naechster Befehl
1620 -;
2000 -;**** modul 2: grafik ein ****
2010 -;
2020 -;legt grafikbildschirm nach $5cØØ
2030 -;und bit-map nach $6ØØØ
2040 -;schaltet in bit-map-modus um
2050 -;und begrenzt basic bis $5cØØ
2060 -;
2070 -han      lda #$7d       ;ort bildschirm und
2080 -          sta vic24      ;bitmap festlegen
2090 -          lda #$c6       ;vic auf abschnitt 1
2100 -          sta cia2       ;richten
2110 -          lda #$5c       ;msb bildschirmstart
2120 -          sta hbase      ;an betriebssystem
2130 -          sta fretop     ;und speicher
2140 -          sta memsiz     ;begrenzen
2150 -          lda vic17      ;bitmapmodus
2160 -          ora #$2Ø       ;einschalten
2170 -          sta vic17
2180 -          rts
3000 -;**** modul 3: bitmap loeschen ****
3010 -;
3020 -;loescht alle bits in der bei
3030 -;$6ØØØ beginnenden bit-map
3040 -;
3050 -gclr     lda #>(bitmap) ;msb bitmapstart
3060 -          sta help+1     ;merken
3070 -          ldy #$ØØ       ;lsb bitmapstart ist null
3080 -          sty help
3090 -          tya            ;nullbytes
3100 -          ldx #$2Ø       ;dez.32 anzahl bloecke
3110 -cloop    sta (help),y   ;byte loeschen
3120 -          iny            ;naechstes byte
3130 -          bne cloop
3140 -          inc help+1     ;msb zeiger erhoehen
3150 -          dex            ;blockzaehler - 1
7160 -          bne cloop
3170 -          rts
4000 -;**** modul 4: farbgebung ****
4010 -;
4020 -;liest aus dem basictext die
4030 -;zeichenfarbe (zf) und die
4040 -;hintergrundfarbe (hf) und belegt
4050 -;den farbspeicher ab $5cØØ mit
4060 -;dieser farbkombination. dabei
4070 -;ist farbe = 16*zf+hf
4080 -;
4090 -color    jsr combyt     ;zf in x-register
4100 -          stx help       ;und merken
4110 -          jsr combyt     ;hf in x-register
4120 -          stx help+1     ;und merken
4130 -;
4140 -          lda help       ;zeichenfarbe
4150 -          asl            ;mal 16
4160 -          asl
4170 -          asl
4180 -          asl
4190 -          clc            ;hf addieren
4200 -          adc help+1
4210 -;
4220 -col2     ldx #>(bild)   ;msb bildschirmadresse
4230 -          stx help+1     ;merken
4240 -          ldy #$ØØ       ;lsb bildschirm ist null
4250 -          sty help       ;merken
4260 -          ldx #$Ø4       ;4 bloecke fuellen
4270 -hloop    sta (help),y   ;farbe eintragen
4280 -          iny            ;naechstes byte
4290 -          bne hloop
4300 -          inc help+1     ;naechster block
4310 -          dex
4320 -          bne hloop
4330 -          rts
5000 -;**** modul 5: textmodus ein ****
5010 -;
5020 -;sorgt dafuer, dass der normale
5030 -;textmodus aktiv wird. grafik
5040 -;bleibt aber erhalten.
5050 -;
5060 -norm     lda #$15       ;normaler bildschirm
5070 -          sta vic24
5080 -          lda #$c7       ;vic auf abschnitt Ø
5090 -          sta cia2
5100 -          lda #$Ø4       ;msb textbildschirm
5110 -          sta hbase      ;an betriebssystem
5120 -          lda vic17      ;textmodus
5130 -          and #$df       ;einschalten
5140 -          sta vic17
5150 -          rts
6000 -;**** modul 6:punkt setzen/loeschen ****
6010 -;
6020 -;setzt in der bitmap ab $6ØØØ
6030 -;einen punkt oder loescht ihn
6040 -;je nach modus. 1.einsprung holt
6050 -;aus basictext parameter x,y,mode
6060 -;2.einsprung setzt gespeicherte
6070 -;parameter voraus.
6080 -;
6090 -;--- parameter holen ----
6100 -;
6110 -punkt1   jsr getnum     ;x und y holen
6120 -          stx yko        ;y merken, x in xko
6130 -          jsr combyt     ;modus holen
6140 -          stx mode       ;und merken
6150 -;
6160 -;--- parameter pruefen ---
6170 -;
6180 -punkt2   lda yko        ;y pruefen
6190 -          cmp #$c7       ;groesser als 200?
6200 -          bcs noplot     ;wenn ja, kein punkt
6210 -          ldy xko+1      ;msb x pruefen
6220 -          beq ok         ;kleiner als 255
6230 -          dey            ;jetzt null?
6240 -          bne noplot     ;msb x > 1,kein punkt
```

```
6250 -          lda xko         ;lsb x pruefen
6260 -          cmp #$4Ø        ;groesser als lsb von 32Ø?
6270 -          bcs noplot      ;dann kein punkt
628Ø -;
629Ø -;--- berechnen byte und bit ---
63ØØ -;
631Ø -ok        lda yko
632Ø -          tax             ;y in akku.und x-register
633Ø -          lsr             ;berechnen y/8
634Ø -          lsr
635Ø -          lsr
636Ø -          tay             ;im y-register merken
637Ø -          clc
638Ø -          txa             ;nochmal y
639Ø -          and #$Ø7        ;(y and 7)
64ØØ -          adc maltab,y    ;+32Ø*y/8 lsb
641Ø -          sta zwsp
642Ø -          lda xko
643Ø -          and #$f8        ;lsb x and 248
644Ø -          adc zwsp
645Ø -          sta zwsp        ;merken lsb
646Ø -          lda maltab1,y   ;32Ø*y/8 msb
647Ø -          adc xko+1
648Ø -          sta zwsp+1      ;msb merken
649Ø -          lda xko         ;bitposition berechnen
65ØØ -          and #$Ø7
651Ø -          tay
652Ø -          lda hochtab,y
653Ø -;
654Ø -;--- mode pruefen ---
655Ø -;
656Ø -          ldy #$ØØ
657Ø -          ldx mode
658Ø -          beq loesch
659Ø -;
66ØØ -;--- punkt setzen ---
661Ø -;
662Ø -          ora (zwsp),y
663Ø -          bne store       ;springt immer
664Ø -;
665Ø -;--- punkt loeschen ---
666Ø -;
667Ø -loesch    eor #$ff        ;loeschen
668Ø -          and (zwsp),y
669Ø -store     sta (zwsp),y    ;eintragen in bitmap
67ØØ -;
671Ø -;--- ruecksprung ---
672Ø -;
673Ø -noplot    rts
7ØØØ -;**** modul 7: linien ziehen ****
7Ø1Ø -;
7Ø2Ø -;zeichnet oder loescht eine linie
7Ø3Ø -;zwischen den punkten (xØ,yØ) und
7Ø4Ø -;(x1,y1). einsprung drwlin holt
7Ø5Ø -;aus basictext die parameter und
7Ø6Ø -;prueft sie. einsprung setpar er-
7Ø7Ø -;wartet die parameter in den
7Ø8Ø -;richtigen speicherstellen xko,
7Ø9Ø -;yko,xk1,yk1,mode. uebergabe an
71ØØ -;modul 6 mittels xko,yko,mode.
711Ø -;
712Ø -ill       rts             ;zurueck bei fehler
713Ø -;
714Ø -;--- einsprung 1 ---
715Ø -;
716Ø -drwlin    jsr gcoord      ;koordinaten 1.punkt holen
717Ø -          bcs ill         ;falscher wert
718Ø -          stx yko         ;yØ merken
719Ø -          sta ct+1        ;msb xØ merken
72ØØ -          sty ct          ;und lsb ebenfalls
721Ø -          jsr gcoord      ;koordinaten 2.punkt holen
722Ø -          bcs ill         ;falscher wert
723Ø -          stx yk1         ;y1 merken
724Ø -          sty xk1         ;lsb x1 merken
725Ø -          ldy ct          ;lsb xØ umladen
726Ø -          sty xko
727Ø -          sta xk1+1       ;msb x1 merken
728Ø -          lda ct+1        ;msb xØ umladen
729Ø -          sta xko+1
73ØØ -          jsr combyt      ;modus holen
731Ø -          stx mode        ;und merken
732Ø -;
733Ø -;--- einsprung 2 ---
734Ø -;
735Ø -setpar    ldy #$Ø1
736Ø -          sty ix          ;inkremente und
737Ø -          sty iy          ;zaehler initialisieren
738Ø -          sty ct
739Ø -          dey
74ØØ -          sty ct+1
741Ø -          sty ax
742Ø -          sty ay
743Ø -          dey
744Ø -;
745Ø -;--- dx berechnen ---
746Ø -;
747Ø -begin     lda xk1+1.      ;ist x1>=xØ?
748Ø -          cmp xko+1       ;msb
749Ø -          bcc drawØ1      ;nein,sprung
75ØØ -          bne drawØ2      ;x1>xØ,sprung
751Ø -          lda xk1         ;beide gleich,lsb pruefen
752Ø -          cmp xko
753Ø -          bcs drawØ2      ;ja,sprung
754Ø -drawØ1    sec             ;dx=xØ-x1
755Ø -          lda xko
756Ø -          sbc xk1
757Ø -          sta dx
758Ø -          lda xko+1
759Ø -          sbc xk1+1
76ØØ -          sta dx+1
761Ø -          sty ix          ;ix ist -1
762Ø -          jmp drawØ3
763Ø -drawØ2    sec             ;dx=x1-xØ
764Ø -          lda xk1
765Ø -          sbc xko
766Ø -          sta dx
767Ø -          lda xk1+1
768Ø -          sbc xko+1
769Ø -          sta dx+1
77ØØ -;
771Ø -;--- berechnen von dy ---
772Ø -;
773Ø -drawØ3    lda yk1         ;ist y1>=yØ?
774Ø -          cmp yko
775Ø -          bcs drawØ4      ;ja,sprung
776Ø -;
777Ø -          sec             ;sonst dy=yØ-y1
778Ø -          lda yko
779Ø -          sbc yk1
78ØØ -          sta dy
781Ø -          sty iy          ;iy ist -1
782Ø -          jmp drawØ5
783Ø -drawØ4    sbc yko         ;dy=y1-yØ
784Ø -          sta dy
785Ø -;
786Ø -drawØ5    lda dx+1        ;dx<dy?
787Ø -          bne drawØ7      ;nein,sprung
788Ø -          lda dx          ;lsb pruefen
789Ø -          cmp dy
79ØØ -          bcs drawØ7      ;dx>=dy,sprung
791Ø -          ldx dy          ;achsen tauschen
792Ø -          sta dy          ;dx nach dy
793Ø -          stx dx          ;dy nach dx
794Ø -          lda ix          ;ay=ix
795Ø -          sta ay
796Ø -          lda iy          ;ax=iy
797Ø -          sta ax
798Ø -          iny
799Ø -          sty ix          ;ix=Ø
8ØØØ -          sty iy          ;iy=Ø
8Ø1Ø -;
8Ø2Ø -drawØ7    lda dx+1        ;of=dx/2
8Ø3Ø -          lsr
8Ø4Ø -          sta of+1        ;ist also Ø
8Ø5Ø -          lda dx
8Ø6Ø -          ror
8Ø7Ø -          sta of
8Ø8Ø -          jmp plotit      ;1.punkt setzen
8Ø9Ø -;
81ØØ -;--- approximation ---
811Ø -;
812Ø -drwlop    lda ix          ;ist ix=-1?
813Ø -          bmi drawØ8      ;ja,sprung
814Ø -          clc             ;xØ=xØ+ix
815Ø -          adc xko
816Ø -          sta xko
817Ø -          lda xko+1
818Ø -          adc #$ØØ
819Ø -          sta xko+1
82ØØ -          jmp draw11
821Ø -drawØ8    sec             ;xØ=xØ-ix
822Ø -          lda xko
823Ø -          sbc #$Ø1
824Ø -          sta xko
825Ø -          lda xko+1
826Ø -          sbc #$ØØ
827Ø -          sta xko+1
828Ø -draw11    clc             ;yØ=yØ+ax
829Ø -          lda yko
83ØØ -          adc ax
831Ø -          sta yko
832Ø -          clc             ;of=of+dy
833Ø -          lda of
834Ø -          adc dy
835Ø -          sta of
836Ø -          lda of+1
837Ø -          adc #$ØØ
838Ø -          sta of+1
839Ø -          inc ct          ;ct=ct+1
84ØØ -          bne drawØ6
841Ø -          inc ct+1
842Ø -drawØ6    lda of+1        ;of<=dx?
843Ø -          cmp dx+1
844Ø -          bcc plotit      ;ja,zeichnen
845Ø -          bne drawØ9      ;groesser,sprung
846Ø -          lda dx
847Ø -          cmp of
848Ø -          bcs plotit      ;ja,zeichnen
849Ø -drawØ9    sec             ;of=of-dx
85ØØ -          lda of
851Ø -          sbc dx
852Ø -          sta of
853Ø -          lda of+1
854Ø -          sbc dx+1
855Ø -          sta of+1
856Ø -          lda ay          ;ist ay=-1?
857Ø -          bmi draw1Ø      ;ja,sprung
858Ø -          clc             ;xØ=xØ+ay
859Ø -          adc xko
86ØØ -          sta xko
861Ø -          lda xko+1
862Ø -          adc #$ØØ
863Ø -          sta xko+1
864Ø -          jmp draw12
865Ø -draw1Ø    sec             ;xØ=xØ-ay
866Ø -          lda xko
867Ø -          sbc #$Ø1
868Ø -          sta xko
869Ø -          lda xko+1
87ØØ -          sbc #$ØØ
871Ø -          sta xko+1
872Ø -draw12    clc             ;yØ=yØ+iy
873Ø -          lda yko
874Ø -          adc iy
875Ø -          sta yko
876Ø -;
877Ø -;--- zeichnen,schleifenende ---
878Ø -;
879Ø -plotit    jsr ok          ;einsprung 3 in modul 6
88ØØ -          lda ct+1        ;ct<=dx?
881Ø -          cmp dx+1
882Ø -          bcc nexpnt      ;ct<dx,sprung
883Ø -          lda dx
884Ø -          cmp ct
885Ø -          bcs nexpnt      ;ct<=dx,sprung
886Ø -          rts             ;linie ist fertig
887Ø -;
888Ø -nexpnt    jmp drwlop      ;naechster punkt
889Ø -;
89ØØ -;--- parameter holen ---
891Ø -;
892Ø -gcoord    jsr chkcom      ;auf komma pruefen
893Ø -          jsr getnum      ;x und y lesen
894Ø -          cpx #$c8        ;ist y>2ØØ?
895Ø -          bcs finish      ;y>=2ØØ,sprung
896Ø -          lda xko+1       ;ist x>32Ø?
897Ø -          cmp #$Ø1        ;msb von x
898Ø -          bcc finis       ;msb ist <1
899Ø -          bne finish      ;msb ist >1
9ØØØ -          ldy xko         ;lsb von x
9Ø1Ø -          cpy #$4Ø        ;lsb von 32Ø
9Ø2Ø -finish    rts             ;wenn carry=1,nach rueckkehr ausstieg bei ill
```

Listing 23. Quelltext im Hypra-Ass-Format zu Hires 4 (Fortsetzung)

```
9030 -finis       ldy xko
9040 -            rts            ;y=xko,a=xko+1
10000-;**** modul 9:kreise zeichnen ****
10010-;
10020-;zeichnet oder loescht einen kreis
10030-;um den mittelpunkt xm,ym und mit
10040-;dem radius r. alle eingaben von
10050-;0 bis maximal 32767 (?) erlaubt.
10060-;beim ersten einsprungspunkt wer-
10070-;den die parameter gelesen und
10080-;geprueft. beim zweiten einsprung
10090-;erwartet programm parameter in
10100-;den richtigen speicherstellen
10110-;xm,ym,r und mode. uebergabe an
10120-;modul 6 mittels xko,yko und mode.
10130-;
10140-err         rts            ;ruecksprung bei fehleingabe
10150-;
10160-;--- parameter holen,pruefen,ablegen ---
10170-;
10180-hole        jsr chkcom     ;komma pruefen
10190-            jsr frmnum     ;xm lesen
10200-            jsr getadr
10210-            lda xko+1      ;msb pruefen
10220-            bmi err        ;zu gross,sprung
10230-            sta xm+1       ;merken
10240-            lda xko        ;lsb
10250-            sta xm         ;merken
10260-;
10270-            jsr chkcom     ;naechstes komma
10280-            jsr frmnum     ;ym lesen
10290-            jsr getadr
10300-            lda xko+1      ;msb pruefen
10310-            bmi err        ;fehleingabe
10320-            sta ym+1       ;merken
10330-            lda xko        ;lsb
10340-            sta ym         ;merken
10350-;
10360-            jsr chkcom     ;komma testen
10370-            jsr frmnum     ;radius lesen
10380-            jsr getadr
10390-            lda xko+1      ;msb pruefen
10400-            bmi err        ;zu gross,sprung
10410-            sta rad+1      ;merken
10420-            lda xko        ;lsb
10430-            sta rad        ;merken
10440-;
10450-            jsr combyt     ;modus holen
10460-            stx mode       ;und merken
10470-;
10480-;--- anfangswerte setzen ---
10490-;
10500-kreis1      lda #$00       ;variable x=0
10510-            sta xko
10520-            sta xko+1
10530-            lda rad        ;variable y=r
10540-            sta help
10550-            lda rad+1
10560-            sta help+1
10570-            sec            ;variable d berechnen:
10580-            lda rad        ;d=r-1
10590-            sbc #$01
10600-            sta ct
10610-            lda rad+1
10620-            sbc #$00
10630-            sta ct+1
10640-;
10650-;--- hauptschleife ---
10660-;
10670-kreislp     lda ct+1       ;ist d<0?
10680-            bpl kreis2     ;nein,sprung
10690-            sec            ;subtraktion
10700-            lda help       ;y=y-1
10710-            sbc #$01
10720-            sta help
10730-            lda help+1
10740-            sbc #$00
10750-            sta help+1
10760-            clc            ;additionen
10770-            lda ct         ;d=d+y+y
10780-            adc help
10790-            sta ct
10800-            lda ct+1
10810-            adc help+1
10820-            sta ct+1
10830-            clc
10840-            lda ct
10850-            adc help
10860-            sta ct
10870-            lda ct+1
10880-            adc help+1
10890-            sta ct+1
10900-;
10910-kreis2      jsr circle     ;up zum pruefen der werte und zeichnen
10920-;
10930-            sec            ;subtraktionen
10940-            lda ct         ;d=d-x-x-1
10950-            sbc xko
10960-            sta ct
10970-            lda ct+1
10980-            sbc xko+1
10990-            sta ct+1
11000-            sec
11010-            lda ct
11020-            sbc xko
11030-            sta ct
11040-            lda ct+1
11050-            sbc xko+1
11060-            sta ct+1
11070-            sec
11080-            lda ct
11090-            sbc #$01
11100-            sta ct
11110-            lda ct+1
11120-            sbc #$00
11130-            sta ct+1
11140-            clc            ;addition x=x+1
11150-            lda xko
11160-            adc #$01
11170-            sta xko
11180-            lda xko+1
11190-            adc #$00
11200-            sta xko+1
11210-;
11220-            cmp help+1     ;ist x<=y
11230-            bcc kreislp    ;ja,sprung
11240-            bne end        ;x>y,kreis fertig
11250-            lda help
11260-            cmp xko
11270-            bcs schleife   ;ja,sprung
11280-end         rts            ;kreis fertig
11290-schleife    jmp kreislp
11300-;
11310-;--- unterprogramm zeichnen ---
11320-;
11330-;berechnet die konkreten punkte
11340-;durch symmetrien, prueft ob die
11350-;sich ergebenden werte fuer x und
11360-;y kleiner als 0 sind und ob y
11370-;groesser als 255. in dem fall
11380-;wird nicht gezeichnet. alle
11390-;anderen pruefungen geschehen im
11400-;modul 6.
11410-;
11420-circle      clc            ;berechne xm+x
11430-            lda xm
11440-            adc xko
11450-            sta xplusx
11460-            lda xm+1
11470-            adc xko+1
11480-            sta xplusx+1
11490-            sec            ;berechne xm-x
11500-            lda xm
11510-            sbc xko
11520-            sta xminx
11530-            lda xm+1
11540-            sbc xko+1
11550-            sta xminx+1
11560-            clc            ;berechne xm+y
11570-            lda xm
11580-            adc help
11590-            sta xplusy
11600-            lda xm+1
11610-            adc help+1
11620-            sta xplusy+1
11630-            sec            ;berechne xm-y
11640-            lda xm
11650-            sbc help
11660-            sta xminy
11670-            lda xm+1
11680-            sbc help+1
11690-            sta xminy+1
11700-            clc            ;berechne ym+y
11710-            lda ym
11720-            adc help
11730-            sta yplusy
11740-            lda ym+1
11750-            adc help+1
11760-            sta yplusy+1
11770-            sec            ;berechne ym-y
11780-            lda ym
11790-            sbc help
11800-            sta yminy
11810-            lda ym+1
11820-            sbc help+1
11830-            sta yminy+1
11840-            clc            ;berechne ym+x
11850-            lda ym
11860-            adc xko
11870-            sta yplusx
11880-            lda ym+1
11890-            adc xko+1
11900-            sta yplusx+1
11910-            sec            ;berechne ym-x
11920-            lda ym
11930-            sbc xko
11940-            sta yminx
11950-            lda ym+1
11960-            sbc xko+1
11970-            sta yminx+1
11980-            lda xko        ;x retten
11990-            pha
12000-            lda xko+1
12010-            pha
12020-            lda xplusx     ;1.punkt
12030-            sta xko
12040-            lda xplusx+1
12050-            sta xko+1
12060-            lda yplusy
12070-            sta yko
12080-            ldy yplusy+1
12090-            jsr setze
12100-            lda yminy      ;2.punkt
12110-            sta yko
12120-            ldy yminy+1
12130-            jsr setze
12140-            lda xminx      ;3.punkt
12150-            sta xko
12160-            lda xminx+1
12170-            ldy yminy+1
12180-            jsr setze
12190-            lda yplusy     ;4.punkt
12200-            sta yko
12210-            ldy yplusy+1
12220-            jsr setze
12230-            lda xplusy     ;5.punkt
12240-            sta xko
12250-            lda xplusy+1
12260-            sta xko+1
12270-            lda yplusx
12280-            sta yko
12290-            ldy yplusx+1
12300-            jsr setze
12310-            lda yminx      ;6.punkt
12320-            sta yko
12330-            ldy yminx+1
12340-            jsr setze
12350-            lda xminy      ;7.punkt
12360-            sta xko
12370-            lda xminy+1
12380-            sta xko+1
12390-            ldy yminx+1
12400-            jsr setze
12410-            lda yplusx     ;8.punkt
12420-            sta yko
12430-            ldy yplusx+1
12440-            jsr setze
12450-            pla            ;xko zurueckholen
12460-            sta xko+1
12470-            pla
12480-            sta xko
12490-            rts            ;zurueck zur hauptschleife
12500-;
12510-;--- pruefen und setzen ---
12520-;
12530-setze       bne fini       ;y>255,kein punkt
12540-            lda xko+1      ;ist x negativ?
12550-            bmi fini       ;ja,kein punkt
12560-            jsr punkt2     ;zum modul 6:punkt setzen
12570-fini        rts            ;weiter in circle
```

```
13000-;**** Modul 12: Tabellen ****
13010-;
13020-;--- Konstanten ---
13030-;
13040-fl199     .by $88,$47,$00,$00,$00;199 im MFLPT-Format
13050-fl319     .by $89,$1f,$80,$00,$00;319 im MFLPT-Format
13060-k199      .by $c7         ;199 als 1-Byte-Integer
13070-;
13080-;--- Variable ---
13090-;
13100-xu        .by $00,$00,$00,$00,$00;XU im MFLPT-Format
13110-yu        .by $00,$00,$00,$00,$00;YU im MFLPT-Format
13120-xo        .by $00,$00,$00,$00,$00;XO im MFLPT-Format
13130-yo        .by $00,$00,$00,$00,$00;YO im MFLPT-Format
13140-befnr     .by $00         ;Befehlsnummer
13150-akku      .by $00         ;Zwischenspeicher fuer Akku-Inhalt
13160-;
13170-;---- Sprungtabelle -------------
13180-;
13190-sprtab    .by $e7
13200-sprtab1   .by $a7
13210-          .wo aus
13220-          .wo han
13230-          .wo gclr
13240-          .wo color
13250-          .wo norm
13260-          .wo punkt1
13270-          .wo drwlin
13280-          .wo hole
13290-          .by 0,0
13300-          .by 0,0
13310-          .by 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13320-;
13330-;---- Befehlstabelle ------------
13340-;
13350-beftab    .tx "aus"
13360-          .by 0
13370-          .tx "han"
13380-          .by 0
13390-          .tx "loe"
13400-          .by 0
13410-          .tx "far"
13420-          .by 0
13430-          .tx "hof"
13440-          .by 0
13450-          .tx "pkt"
13460-          .by 0
13470-          .tx "lin"
13480-          .by 0
13490-          .tx "cir"
13500-          .by 0
13510-          .by 0,0
13520-          .by 0
13530-          .by 0,0
13540-          .by 0,0
13550-;
13560-;--- Platz fuer weitere Befehle ---
13570-;
13580-          .by 0,0,0,0,0,0,0,0,0,0
13590-          .by 0,0,0,0,0,0,0,0,0,0
13600-          .by 0,0,0,0,0,0,0,0,0,0
13610-          .by 0,0,0,0,0,0,0,0,0,0
13620-          .by 0,0,0,0,0,0,0,0,0,0
13630-;
13640-;--- Multiplikationstabellen ---
13650-;
13660-maltab    .by $00,$40,$80,$c0;LSB
13670-          .by $00,$40,$80,$c0
13680-          .by $00,$40,$80,$c0
13690-          .by $00,$40,$80,$c0
13700-          .by $00,$40,$80,$c0
13710-          .by $00,$40,$80,$c0
13720-
13730-;
13740-maltab1   .by $60,$61,$62,$63;MSB
13750-          .by $65,$66,$67,$68
13760-          .by $6a,$6b,$6c,$6d
13770-          .by $6f,$70,$71,$72
13780-          .by $74,$75,$76,$77
13790-          .by $79,$7a,$7b,$7c
13800-          .by $7e
13810-;
13820-hochtab   .by $80,$40,$20,$10;Zweierpotenzen
13830-          .by $08,$04,$02,$01
13840-;
13850-test
13860-;


ready.
```

Listing 23. Quelltext im Hypra-Ass-Format zu Hires 4 (Schluß)

```
Name : hires4.obj          c000 c505
-----------------------------------
c000 : a9 16 8d 08 03 a9 c0 8d   b4
c008 : 09 03 60 a9 e4 8d 08 03   c1
c010 : a9 a7 8d 09 03 60 20 73   ac
c018 : 00 c9 60 b0 19 c9 41 90   31
c020 : 15 8d 4a c4 a2 00 8e 49   1e
c028 : c4 a0 00 ee 49 c4 bd 71   af
c030 : c4 d0 09 ad 4a c4 20 79   93
c038 : 00 4c e7 a7 d1 7a d0 28   d2
c040 : c8 e8 bd 71 c4 d0 f5 18   f5
c048 : 98 65 7a 85 7a 90 02 e6   e4
c050 : 7b ad 49 c4 0a aa bd 4b   10
c058 : c4 8d 63 c0 bd 4c c4 8d   40
c060 : 64 c0 20 ff ff 4c 36 c0   e9
c068 : e8 bd 71 c4 d0 fa e8 4c   45
c070 : 29 c0 a9 7d 8d 18 d0 a9   44
c078 : c6 8d 00 dd a9 5c 8d 88   85
c080 : 02 85 34 85 38 ad 11 d0   d9
c088 : 09 20 8d 11 d0 60 a9 60   9e
c090 : 85 fc a0 00 84 fb 98 a2   8b
c098 : 20 91 fb c8 d0 fb e6 fc   1b
c0a0 : ca d0 f6 60 20 00 e2 86   37
c0a8 : fb 20 00 e2 86 fc a5 fb   ef
c0b0 : 0a 0a 0a 0a 18 65 fc a2   69
c0b8 : 5c 86 fc a0 00 84 fb a2   04
c0c0 : 04 91 fb c8 d0 fb e6 fc   27
c0c8 : ca d0 f6 60 a9 15 8d 18   6e
c0d0 : d0 a9 c7 8d 00 dd a9 04   b6
c0d8 : 8d 88 02 ad 11 d0 29 df   dc
c0e0 : 8d 11 d0 60 20 eb b7 86   83
c0e8 : 09 20 00 e2 86 02 a5 09   7f
c0f0 : c9 c7 b0 44 a4 15 f0 09   1a
c0f8 : 88 d0 3d a5 14 c9 40 b0   de
c100 : 37 a5 09 aa 4a 4a 4a a8   13
c108 : 18 8a 29 07 79 ca c4 85   9d
c110 : fd a5 14 29 f8 65 fd 85   c8
c118 : fd b9 e3 c4 65 15 85 fe   96
c120 : a5 14 29 07 a8 b9 fc c4   d0
c128 : a0 00 a6 02 f0 04 11 fd   22
c130 : d0 04 49 ff 31 fd 91 fd   9a
c138 : 60 60 20 67 c2 b0 fa 86   68
c140 : 09 85 94 84 93 20 67 c2   1f
c148 : b0 ef 86 0a 84 8b a4 93   31
c150 : 84 14 85 8c a5 94 85 15   10
c158 : 20 00 e2 86 02 a0 01 84   34
c160 : 1c 84 19 84 93 88 84 94   4e
c168 : 84 1b 84 1a 88 a5 8c c5   52
c170 : 15 90 08 d0 18 a5 8b c5   52
c178 : 14 b0 12 38 a5 14 e5 8b   1a
c180 : 85 fb a5 15 e5 8c 85 fc   e2
c188 : 84 1c 4c 9a c1 38 a5 8b   0c
c190 : e5 14 85 fb a5 8c e5 15   e1
c198 : 85 fc a5 0a c5 09 b0 0c   c5
c1a0 : 38 a5 09 e5 0a 85 11 84   c4
c1a8 : 19 4c b0 c1 e5 09 85 11   2a
c1b0 : a5 fc d0 19 a5 fb c5 11   9e
c1b8 : b0 13 a6 11 85 11 86 fb   b1
c1c0 : a5 1c 85 1a a5 19 85 1b   87
c1c8 : c8 84 1c 84 19 a5 fc 4a   b1
c1d0 : 85 8c a5 fb 6a 85 8b 4c   1e
c1d8 : 54 c2 a5 1c 30 0e 18 65   19
c1e0 : 14 85 14 a5 15 69 00 85   18
c1e8 : 15 4c f9 c1 38 a5 14 e9   af
c1f0 : 01 85 14 a5 15 e9 00 85   19
c1f8 : 15 18 a5 09 65 1b 85 09   fb
c200 : 18 a5 8b 65 11 85 8b a5   31
c208 : 8c 69 00 85 8c e6 93 d0   e9
c210 : 02 e6 94 a5 8c c5 fc 90   6b
c218 : 3b d0 06 a5 fb c5 8b b0   6f
c220 : 33 38 a5 8b e5 fb 85 8b   b5
c228 : a5 8c e5 fc 85 8c a5 1a   b4
c230 : 30 0e 18 65 14 85 14 a5   23
c238 : 15 69 00 85 15 4c 4d c2   21
c240 : 38 a5 14 e9 01 85 14 a5   65
c248 : 15 e9 00 85 15 18 a5 09   bd
c250 : 65 19 85 09 20 01 c1 a5   21
c258 : 94 c5 fc 90 07 a5 fb c5   39
c260 : 93 b0 01 60 4c da c1 20   7b
c268 : fd ae 20 eb b7 e0 c8 b0   49
c270 : 0c a5 15 c9 01 90 07 d0   20
c278 : 04 a4 14 c0 40 60 a4 14   ad
c280 : 60 60 20 fd ae 20 8a ad   4a
c288 : 20 f7 b7 a5 15 30 f2 85   f0
c290 : 8c a5 14 85 8b 20 fd ae   b4
c298 : 20 8a ad 20 f7 b7 a5 15   6b
c2a0 : 30 df 85 8e a5 14 85 8d   1f
c2a8 : 20 fd ae 20 8a ad 20 f7   fd
c2b0 : b7 a5 15 30 cc 85 96 a5   24
c2b8 : 14 85 95 20 00 e2 86 02   2d
c2c0 : a9 00 85 14 85 15 a5 95   10
c2c8 : 85 fb a5 96 85 fc 38 a5   f3
c2d0 : 95 e9 01 85 93 a5 96 e9   df
c2d8 : 00 85 94 a5 94 10 27 38   4b
c2e0 : a5 fb e9 01 85 fb a5 fc   e6
c2e8 : e9 00 85 fc 18 a5 93 65   9a
c2f0 : fb 85 93 a5 94 65 fc 85   bb
c2f8 : 94 18 a5 93 65 fb 85 93   e8
c300 : a5 94 65 fc 85 94 20 4d   00
c308 : c3 38 a5 93 e5 14 85 93   ff
c310 : a5 94 e5 15 85 94 38 a5   44
c318 : 93 e5 14 85 93 a5 94 e5   d8
c320 : 15 85 94 38 a5 93 e9 01   c5
c328 : 85 93 a5 94 e9 00 85 94   51
c330 : 18 a5 14 69 01 85 14 a5   25
c338 : 15 69 00 85 15 c5 fc 90   47
c340 : 9a d0 06 a5 fb c5 14 b0   18
c348 : 01 60 4c db c2 18 a5 8b   a2
c350 : 65 14 85 5f a5 8c 65 15   8b
c358 : 85 60 38 a5 8b e5 14 85   13
c360 : 9b a5 8c e5 15 85 9c 18   ce
c368 : a5 8b 65 fb 85 9e a5 8c   a9
c370 : 65 fc 85 9f 38 a5 8b e5   53
c378 : fb 85 a3 a5 8c e5 fc 85   ca
c380 : a4 18 a5 8d 65 fb 85 57   46
c388 : a5 8e 65 fc 85 58 38 a5   b4
c390 : 8d e5 fb 85 59 a5 8e e5   88
c398 : fc 85 5a 18 a5 8d 65 14   75
c3a0 : 85 5b a5 8e 65 15 85 5c   dc
c3a8 : 38 a5 8d e5 14 85 5d a5   01
```

Listing 24. »Hires 4.OBJ« ist der Maschinencode zum Listing 23. Das Programm ist mit dem MSE einzugeben.

```
c3b0 : 8e e5 15 85 5e a5 14 48   1b
c3b8 : a5 15 48 a5 5f 85 14 a5   6c
c3c0 : 60 85 15 a5 57 85 09 a4   ec
c3c8 : 58 20 20 c4 a5 59 85 09   1e
c3d0 : a4 5a 20 20 c4 a5 9b 85   a0
c3d8 : 14 a5 9c a4 5a 20 20 c4   2b
c3e0 : a5 57 85 09 a4 58 20 20   81
c3e8 : c4 a5 9e 85 14 a5 9f 85   cf
c3f0 : 15 a5 5b 85 09 a4 5c 20   c7
c3f8 : 20 c4 a5 5d 85 09 a4 5e   7f
c400 : 20 20 c4 a5 a3 85 14 a5   18
c408 : a4 85 15 a4 5e 20 20 c4   3a
c410 : a5 5b 85 09 a4 5c 20 20   d3
c418 : c4 68 85 15 68 85 14 60   d8
c420 : d0 07 a5 15 30 03 20 ee   f9
c428 : c0 60 88 47 00 00 00 89   36
c430 : 1f 80 00 00 c7 00 00 00   0c
c438 : 00 00 00 00 00 00 00 00   39
c440 : 00 00 00 00 00 00 00 00   41
c448 : 00 00 00 e7 a7 0b c0 72   00
c450 : c0 8e c0 a4 c0 cc c0 e4   5b
c458 : c0 3a c1 82 c2 00 00 00   22
c460 : 00 00 00 00 00 00 00 00   61
c468 : 00 00 00 00 00 00 00 00   69
c470 : 00 41 55 53 00 48 41 4e   b5
c478 : 00 4c 4f 45 00 46 41 52   f7
c480 : 00 48 4f 46 00 50 4b 54   99
c488 : 00 4c 49 4e 00 43 49 52   ae
c490 : 00 00 00 00 00 00 00 00   91
c498 : 00 00 00 00 00 00 00 00   99
c4a0 : 00 00 00 00 00 00 00 00   a1
c4a8 : 00 00 00 00 00 00 00 00   a9
c4b0 : 00 00 00 00 00 00 00 00   b1
c4b8 : 00 00 00 00 00 00 00 00   b9
c4c0 : 00 00 00 00 00 00 00 00   c1
c4c8 : 00 00 00 40 80 c0 00 40   5f
c4d0 : 80 c0 00 40 80 c0 00 40   47
c4d8 : 80 c0 00 40 80 c0 00 40   4f
c4e0 : 80 c0 00 60 61 62 63 65   4e
c4e8 : 66 67 68 6a 6b 6c 6d 6f   18
c4f0 : 70 71 72 74 75 76 77 79   20
c4f8 : 7a 7b 7c 7e 80 40 20 10   c9
c500 : 08 04 02 01 c3 ff 00 00   e7
```

Listing 24. »Hires 4.OBJ« (Schluß)

```
10 REM ********************************       <060>
20 REM *                              *       <069>
30 REM *      H I R E S 4 . B A S     *       <029>
40 REM *  BASICVERSION EINES GRAFIK-  *       <004>
50 REM *  PROGRAMMES FUER DEN C64     *       <051>
60 REM *   HEIMO PONNATH  HAMBURG     *       <104>
70 REM *                              *       <119>
80 REM ********************************       <130>
90 REM                                        <152>
100 REM ------- HAUPTPROGRAMM ---------       <242>
110 PRINT CHR$(147)                           <139>
120 PRINT:PRINT:PRINT"DIESES PROGRAMM TEST
    ET ALLE MODULE":PRINT"VON HIRES4.BAS"     <026>
130 ZF=5:HF=0:REM ZEICHEN- UND HINTERGRUND
    FARBE                                     <176>
140 GOSUB 2000:REM GRAFIK EINSCHALTEN         <012>
150 GOSUB 3000:REM BITMAP LOESCHEN            <166>
160 GOSUB 4000:REM FARBGEBUNG                 <231>
170 L=0:REM LOESCHMARKE AUF PUNKT SETZEN S
    TELLEN                                    <250>
180 FOR X=0 TO 319                            <018>
190 :Y=50*SIN(X/20)+100                       <153>
200 :GOSUB 6000:REM PUNKTE SETZEN             <014>
210 NEXT X                                    <158>
220 L=1:REM LOESCHMARKE AUF PUNKT LOESCHEN
     STELLEN                                  <089>
230 FOR X=100 TO 200                          <108>
240 :Y=50*SIN(X/20)+100                       <203>
250 :GOSUB 6000:REM PUNKTE LOESCHEN           <113>
260 NEXT X                                    <210>
270 GOSUB 5000:REM TEXTMODUS EINSCHALTEN      <230>
280 PRINT:PRINT"DIE MODULE 2 BIS 6 SIND OK
    "                                         <187>
290 FOR I=0 TO 1000:NEXT I                    <216>
300 GOSUB 2000:REM GRAFIK WIEDER EINSCHALT
    EN                                        <191>
310 X1=160:Y1=100:R%=90:L=0                   <180>
320 FOR A=0 TO 6.28 STEP .05                  <187>
330 :X2=R%*COS(A)+160                         <230>
340 :Y2=R%*SIN(A)+100                         <247>
350 :GOSUB 7000:REM LINIEN ZEICHNEN           <146>
360 NEXT A                                    <126>
370 GOSUB 5000:REM TEXTMODUS EINSCHALTEN      <074>
380 PRINT:PRINT"AUCH DAS MODUL 7 ARBEITET"    <194>
390 FOR I=0 TO 1000:NEXT I                    <060>
400 GOSUB 2000:REM GRAFIK EINSCHALTEN         <018>
410 XM=160:YM=100:R=95:L=0                    <137>
420 GOSUB 8000:REM KREIS ZEICHNEN             <028>
430 FOR I=0 TO 1000:NEXT I                    <100>
440 GOSUB 5000:REM TEXTMODUS                  <183>
450 PRINT:PRINT"AUCH DAS MODUL 8 FUNKTIONI
    ERT."                                     <002>
460 POKE 52,160:POKE 56,160:REM BASICZEIGE
    R ZURUECKSTELLEN                          <207>
470 CLR                                       <074>
480 END                                       <228>
2000 REM - MODUL 2:GRAFIK EINSCHALTEN -       <089>
2010 POKE 56576,198 : REM ABSCHNITT 1 EINS
     TELLEN                                   <000>
2020 POKE 53272,125 : REM BILDSCHIRM UND B
     ITMAPSTART FESTLEGEN                     <158>
2030 POKE 648,92     : REM BILDSCHIRM AN BE
     TRIEBSSYSTEM MELDEN                      <017>
2040 POKE 52,92      : REM BASICSPEICHER BE
     GRENZEN                                  <070>
2050 POKE 56,92                               <104>
2060 POKE 53265,PEEK(53265) OR 32:REM BITM
     APMODUS EINSCHALTEN                      <055>
2070 RETURN                                   <096>
3000 REM - MODUL 3: BITMAP LOESCHEN ---       <015>
3010 BA=24576 : REM BITMAPSTARTADRESSE        <105>
3020 FOR I=BA TO BA+7999 : REM DIE LOESCHS
     CHLEIFE                                  <172>
3030 :POKE I,0                                <181>
3040 NEXT I                                   <074>
3050 RETURN                                   <058>
4000 REM - MODUL 4: FARBGEBUNG --------       <060>
4010 BS=23552 : REM BILDSCHIRMSTARTADRESSE    <118>
4020 F=16*ZF+HF:REM BERECHNUNG DES FARBCOD
     ES                                       <182>
4030 FOR I=BS TO BS+999 : REM DIE FARBSCHL
     EIFE                                     <190>
4040 :POKE I,F                                <052>
4050 NEXT I                                   <068>
4060 RETURN                                   <052>
5000 REM - MODUL 5: TEXTMODUS EIN -----       <055>
5010 POKE 53272,21 : REM ALTEN INHALT REST
     AURIEREN                                 <050>
5020 POKE 56576,199: REM VIC WIEDER AUF AB
     SCHNITT 0                                <179>
5030 POKE 648,4 :REM ALTEN BILDSCHIRM AN B
     ETRIEBSSYSTEM MELDEN                     <098>
5040 POKE 53265,PEEK(53265) AND 223:REM TE
     XTMODUS EINSCHALTEN                      <029>
5050 RETURN                                   <026>
6000 REM - MODUL 6:PUNKT SETZEN/LOESCHEN      <006>
6010 IF X<0 OR X>319 OR Y<0 OR Y>199 THEN
     6060                                     <218>
6020 BY=(X AND 504)+40*(Y AND 248)+(Y AND
     7)                                       <215>
6030 BI=7 - (X AND 7)                         <250>
6040 IF L=1 THEN POKE BY+BA,PEEK(BY+BA) AN
     D NOT (2↑BI):GOTO 6060                   <226>
6050 POKE BY+BA,PEEK(BY+BA) OR (2↑BI)         <170>
6060 RETURN                                   <020>
7000 REM - MODUL 7: LINIEN ZEICHNEN/LOESCH
     EN -                                     <115>
7010 X%=X1:Y%=Y1:REM ALLES IN INTEGERS        <242>
7020 DX%=ABS(X2-X1):REM BERECHNUNG DER DIF
     FERENZEN                                 <115>
7030 DY%=ABS(Y2-Y1)                           <093>
7040 S1%=SGN(X2-X1):REM BESTIMMUNG DER OKT
     ANTEN                                    <128>
7050 S2%=SGN(Y2-Y1)                           <244>
7060 IF DX%<DY% THEN D%=DX%:DX%=DY%:DY%=D%
     :T%=1:GOTO 7080                          <235>
7070 T%=0 : REM UMDREHEN WENN DX < DY UND
     MARKE SETZEN                             <092>
7080 D%=2*DY% - DX%:REM STARTWERT D(1) ERM
     ITTELN                                   <116>
7090 I1%=2*DY% : REM INKREMENTE BERECHNEN     <204>
7100 I2%=2*DX%                                <186>
7110 FOR I=1 TO DX% : REM HAUPTSCHLEIFE       <134>
7120 :X=X%:Y=Y%:GOSUB 6000:REM PUNKT ZEICH
     NEN                                      <147>
7130 :IF D%<0 THEN GOTO 7170                  <216>
```

```
7140 :IF T%=1 THEN X%=X%+S1%:GOTO 7160         <116>
7150 :Y%=Y%+S2%                                 <167>
7160 :D%=D%-I2%                                 <177>
7170 :IF T%=1 THEN Y%=Y%+S2%:GOTO 7190         <104>
7180 :X%=X%+S1%                                 <123>
7190 :D%=D%+I1%                                 <204>
7200 NEXT I                                     <070>
7210 RETURN                                     <156>
8000 REM - MODUL 8: KREIS ZEICHNEN/LOESCHE
     N -                                        <114>
8010 X%=0:Y%=R:REM STARTWERTE                   <185>
8020 D%=R-1:REM ENTSCHEIDUNGSGROESSE            <151>
8030 IF D%<0 THEN Y%=Y%-1:D%=D%+Y%+Y%           <243>
8040 GOSUB 8100:REM ZEICHNEN VON 8 KREISPU
     NKTEN                                      <234>
8050 D%=D%-X%-X%-1                              <224>
8060 X%=X%+1                                    <073>
8070 IF X%<=Y% THEN 8030                        <226>
8080 RETURN                                     <008>
8100 REM - UP 8 KREISPUNKTE ZEICHNEN -          <088>
8110 X=XM+X%:Y=YM+Y%:GOSUB 6000                 <162>
8120 X=XM-X%:Y=YM+Y%:GOSUB 6000                 <236>
8130 X=XM+X%:Y=YM-Y%:GOSUB 6000                 <246>
8140 X=XM-X%:Y=YM-Y%:GOSUB 6000                 <064>
8150 X=XM+Y%:Y=YM+X%:GOSUB 6000                 <202>
8160 X=XM-Y%:Y=YM+X%:GOSUB 6000                 <020>
8170 X=XM+Y%:Y=YM-X%:GOSUB 6000                 <030>
8180 X=XM-Y%:Y=YM-X%:GOSUB 6000                 <104>
8190 RETURN                                     <118>
```

Listing 25. Die Basic-Version zu »Hires 4« enthält neben allen erforderlichen Unterroutinen ein kleines Testprogramm

```
10 REM ********************************         <060>
20 REM *                              *         <069>
30 REM *      H I R E S 4 . A S S     *         <054>
40 REM *  TEST DER GRAFIKERWEITERUNG  *         <146>
50 REM *     HIRES 4  FUER DEN C64    *         <098>
60 REM *    HEIMO PONNATH  HAMBURG    *         <104>
70 REM *                              *         <119>
80 REM ********************************         <130>
90 REM                                          <152>
100 REM ------- HAUPTPROGRAMM ---------         <242>
110 PRINT CHR$(147)                             <139>
120 PRINT:PRINT:PRINT"DIESES PROGRAMM TEST
    ET ALLE MODULE":PRINT"VON HIRES 4"          <137>
130 ZF=5:HF=0:REM ZEICHEN- UND HINTERGRUND
    FARBE                                       <176>
140 SYS 49152:HAN:REM GRAFIK EINSCHALTEN        <084>
150 LOE:REM BITMAP LOESCHEN                     <027>
160 FAR,ZF,HF:REM FARBGEBUNG                    <235>
170 L=1:REM LOESCHMARKE AUF PUNKT SETZEN S
    TELLEN                                      <010>
180 FOR X=0 TO 319                              <018>
190 :Y=50*SIN(X/20)+100                         <153>
200 :PKT X,Y,L:REM PUNKTE SETZEN                <211>
210 NEXT X                                      <158>
220 L=0:REM LOESCHMARKE AUF PUNKT LOESCHEN
    STELLEN                                     <073>
230 FOR X=100 TO 200                            <108>
240 :Y=50*SIN(X/20)+100                         <203>
250 :PKT X,Y,L:REM PUNKTE LOESCHEN              <133>
260 NEXT X                                      <210>
270 HOF:REM TEXTMODUS EINSCHALTEN               <001>
280 PRINT:PRINT"DIE MODULE 2 BIS 6 SIND OK
    "                                           <187>
290 FOR I=0 TO 1000:NEXT I                      <216>
300 HAN:REM GRAFIK WIEDER EINSCHALTEN           <160>
310 X1=160:Y1=100:R%=90:L=1                     <181>
320 FOR A=0 TO 6.28 STEP .05                    <187>
330 :X2=R%*COS(A)+160                           <230>
340 :Y2=R%*SIN(A)+100                           <247>
350 :LIN,X1,Y1,X2,Y2,L:REM LINIEN ZEICHNEN      <080>
360 NEXT A                                      <126>
370 HOF:REM TEXTMODUS EINSCHALTEN               <101>
380 PRINT:PRINT"AUCH DAS MODUL 7 ARBEITET"      <194>
390 FOR I=0 TO 1000:NEXT I                      <060>
400 HAN:REM GRAFIK EINSCHALTEN                  <068>
410 XM=160:YM=100:R=95:L=1                      <009>
420 CIR,XM,YM,R,L:REM KREIS ZEICHNEN            <254>
430 FOR I=0 TO 1000:NEXT I                      <100>
440 HOF:AUS:REM TEXTMODUS UND ERWEITERUNG
    ABSCHALTEN                                  <153>
450 PRINT:PRINT"AUCH DAS MODUL 8 FUNKTIONI
    ERT."                                       <002>
460 POKE 52,160:POKE 56,160:REM BASICZEIGE
    R ZURUECKSTELLEN                            <207>
470 CLR                                         <074>
480 END                                         <228>
```

Listing 26. Dieses Programm ist identisch zum Testprogramm in Listing 25. Allerdings werden hier die Befehle mit der Erweiterung »Hires 4« bearbeitet.

Einstieg in die dritte Dimension

In diesem Abschnitt unseres Grafik-Kurses entschleiern wir die wenigen mathematischen Notwendigkeiten, die für die anspruchsvolle grafische Datenverarbeitung unerläßlich sind. Dabei ist das mathematische Rüstzeug alles andere als schwierig: Sogar zum Verstehen der unumgänglichen Matrizenrechnung genügt die Kenntnis der vier Grundrechenarten (plus, minus, mal und geteilt) und die konsequente Anwendung von einigen wenigen mathematischen Regeln.

Wir werden uns auf möglichst einfache Weise mit der Matrizenrechnung vertraut machen und dann einen kurzen Streifzug durch die Grundlagen der 2D- und der 3D-Grafik unternehmen.

Das Bildschirmkoordinatensystem

Fangen wir mit einer Bestandsaufnahme an. Letztendlich müssen wir alle grafischen Kunstwerke, die wir produzieren, auf unserem Bildschirm darstellen (von anderen Grafik-Ausgabegeräten wollen wir hier einmal absehen). Dabei haben wir eine horizontale Ausdehnung von 320 Punkten und eine vertikale von 200. Die Numerierung in der X-Richtung läuft von 0 bis 319 nach rechts und die in der Y-Richtung von 0 bis 199 nach unten.

Es handelt sich in diesem Fall um ein sogenanntes linkshändiges Koordinatensystem. Linkshändig deswegen, weil man beim Drehen der X-Achse um den Ursprung auf dem kürzesten Weg zur Y-Achse die Richtung einschlägt, in die die Finger der linken Hand bei offener Faust zeigen, wenn der Daumen dabei aus der Bildschirmebene herausweist. Das ist ein etwas ungewöhnliches System: Unser Denken spielt sich im allgemeinen in rechtshändigen Koordinatensystemen ab. So setzen wir beispielsweise immer dann, wenn von der Höhe eines Turmes die Rede ist, voraus, daß er von unten nach oben gemessen wurde, denn alle Höhenangaben in Landkarten sind normalerweise in »Höhe über Normalnull« gemacht.

Weltkoordinaten

Damit wären wir bei den sogenannten Weltkoordinaten gelandet: Man versteht darunter Koordinatensysteme, die einer bestimmten realen Situation gerecht werden.

So könnte man nun beispielsweise den Ursprung eines Koordinatensystems in den Fußpunkt der Cheopspyramide legen, dessen Maßeinheit auf beiden Achsen jeweils 1 Meter beträgt. Dann läge die Spitze der Pyramide im Punkt S(0,137) und in der Waagrechten befände sich der letzte Stein, der zur Pyramide gehört, etwa im Punkt P(115,0).

Ein Biologe, der den Tabakmosaikvirus untersucht, könnte ein Koordinatensystem gebrauchen, dessen Maßeinheit ein Mikrometer ist. Wenn er dann den Ursprung an ein Ende dieses Virus legt, dann würde durch eine Linie von P1(0,0) nach P2(0,0.04) die Breite des stabförmigen Virus durchmessen werden, die Länge aber durch eine Strecke von P1(0,0) nach P3(17,0).

Transformationen

Sie sehen vielleicht schon, daß man beliebig viele Weltkoordinatensysteme definieren kann. Wie kann nun der besagte Biologe sein Untersuchungsobjekt möglichst bildfüllend auf dem Bildschirm zeigen? Oder wie könnte man die Cheopspyramide als Computergrafik präsentieren?

Diese Aufgabe zu lösen, ist durch sogenannte Transformationen möglich. Der Begriff stammt vom spätlateinischen »transformatio«, was eine Umbildung oder Verwandlung von etwas bedeutet, ohne dessen Wert zu verändern. Eine Transformation ist also eine Kette von Rechnungen, ein Algorithmus, der jeden Wert aus einem Weltkoordinatensystem umsetzt in einen unseres Bildschirmsystems. Wenn unser Biologe beispielsweise ein Weltkoordinatensystem benutzt, das in der Horizontalen von 0 bis 32 Mikrometer und in der Vertikalen von 0 bis 20 Mikrometer reicht, dann sorgt die Transformation für eine Anpassung dieser Ausdehnung an die Fläche des Bildschirmes. Meist ist man bemüht, außerdem noch aus der Linkshändigkeit des Bildschirmsystems eine Rechtshändigkeit zu machen: Auch das ist die Aufgabe einer Transformation.

Noch offensichtlicher wird die Notwendigkeit von Transformationen bei der Darstellung von räumlichen Objekten auf dem Bildschirm. Unser Bildschirm erlaubt nur flächige Bilder. Es gibt keine Möglichkeit, einen Punkt vor oder hinter die Oberfläche der Fernsehröhre zu plazieren. 3D-Gebilde müssen also umgewandelt werden in 2D-Objekte, müssen irgendwie projiziert werden – ähnlich dem Schattenriß eines räumlichen Gebildes, das in den Lichtstrahl eines Diaprojektors gerät und auf der Fläche der Leinwand sein 2D-Abbild zeigt. Auch solche Projektionen werden durch Transformationen gemeistert.

Nicht nur beim Übergang von Weltsystemen flächiger oder räumlicher Art zum Bildschirmsystem, sondern auch innerhalb eines Systems bedient man sich der Transformationen. Das Verkleinern, Vergrößern, Verschieben oder Rotieren von grafischen Objekten erfolgt durch rechnerische Übergänge zwischen verschiedenen Koordinatensystemen. Jeder dieser häufig gebrauchten Übergänge ist eine Transformation.

Matrixrechnung

Wir haben nun so oft von Transformationen gesprochen, daß Sie sicherlich neugierig geworden sind, wie das nun vor sich geht. Wir erwähnten vorhin schon, daß eine Transformation aus einer Reihe von Rechenschritten besteht. Jeder dieser Rechenschritte wird auf jeden bildwichtigen Punkt im Ausgangssystem angewendet, um als Ergebnis diesen Punkt im Endsystem zu erhalten. Besonders bei komplexeren Transformationen ergäbe sich auf diese Weise eine derart große Anzahl von Rechnungen, daß man sie kaum überschauen könnte. Und vor allem: Die Rechenzeit im Computer würde nahezu untragbar. Hier hilft uns nun die Matrix aus der Klemme. Es ist nämlich möglich, für jede auch noch so komplexe Folge von Transformationen eine einzige sogenannte Transformationsmatrix zu berechnen und damit die Matrix aller bildwichtigen Punkte zu multiplizieren. Alles läßt sich also auf eine einzige Matrizenmultiplikation zurückführen, die relativ einfach und schnell durch ein Programm erledigt werden kann. Die Hauptaufgabe des Programmierers besteht darin, für seine Aufgabenstellung die geeignete Transformationsmatrix zu erstellen. Die verschiedenen Möglichkeiten solcher Matrizen lernen Sie in diesem Teil kennen. Was aber ist denn nun eigentlich eine Matrix?

James Joseph Sylvester (1814 – 1897) führte 1850 den Begriff der Matrix ein. Lateinisch »mater« heißt »Mutter«, französisch »matrice« bedeutet auch »Gußform«. Sylvester wollte mit dieser Bezeichnung wohl eine häufige Verwendung von Matrizen charakterisieren: Dabei dient die Matrix gewissermaßen als eine Gußform, durch die Daten in gewisse neue Formen und Zusammenhänge gebracht werden können.

Genug von Geschichte! Wo können Matrizen überall angewendet werden (außer in der Computergrafik, die unser Anliegen ist)? Dem Techniker und Ingenieur dienen sie beispielsweise zur Ermittlung von Eigenfrequenzen in der Schwingungstechnik, zu Netzberechnungen in der Elektrotechnik oder zur Berechnung statisch unbestimmter Systeme in der Baustatik. Der Physiker bedient sich ihrer in der Quantentheorie. Kaufleute, Betriebs- und Volkswirte erleichtern sich die Produktionsplanung, Materialplanung und Betriebskostenüberwachung damit. Matrizen sind aber auch Handwerkszeuge für die einfache Erfassung von komplexen Zusammenhängen: Man kann damit Verflechtungsbilanzen erstellen und untersuchen. Sehr interessant scheinen auch die vielfältigen Möglichkeiten bei Optimierungsproblemen zu sein. Jetzt müßte Ihnen deutlich geworden sein, welch ein breites Anwendungsspektrum sich da offenbart. Allen ist eines gemeinsam: Es liegen sogenannte Vielfaktorenprobleme vor. Damit ist gemeint, daß man eine große Anzahl von Einflußgrößen rechnerisch zu bewältigen hat, und das geht halt mit Matrizen sehr einfach und computergerecht.

Was sind Matrizen?

Nun ist der Ausdruck Matrix schon so oft gefallen und Sie sollen endlich erfahren, was das eigentlich ist. Eine Matrix ist eine geordnete rechteckige Darstellung von Elementen. Elemente können Zahlen sein oder Formeln oder auch Texte. Bild 27 zeigt Ihnen ein Beispiel einer Matrix, deren Elemente Zahlen sind.

$$A = \begin{bmatrix} 20 & 1 & 150 \\ 2 & 2 & 400 \\ 15 & 1 & 100 \\ 12 & 0 & 80 \end{bmatrix}$$

Bild 27. Beispiel einer Matrix mit Zahlen als Elementen

Es gibt verschiedene Schreibweisen für Matrizen. Die in Bild 27 – mit eckigen Klammern und einem Großbuchstaben als Namen der Matrix – ist weit verbreitet, und wir werden sie im folgenden verwenden. Unsere Matrix hat vier Zeilen und drei Spalten. Man spricht dann von einer 4,3-Matrix. Will man nicht eine ganz konkrete, sondern eine allgemeine 4,3-Matrix angeben, dann verwendet man anstelle der Zahlen – wie allgemein in der Mathematik – Buch-

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

Bild 28. Eine allgemeine 4,3-Matrix

staben, an die Indices gehängt sind. In Bild 28 ist unsere Beispiel-Matrix auf diese Weise angegeben.

Der erste Index ist dann immer die Zeilennummer, der zweite die Spaltennummer. Jetzt wissen Sie zwar, was eine Matrix ist, aber was man konkret mit diesen Gebilden anfangen kann, ist Ihnen vermutlich noch unklar. Sehen wir uns daher ein praktisches Beispiel an.

Matrizen und Tabellen

Matrizen kann man in vielen Fällen einfach als das mathematische Gegenstück zu Tabellen auffassen. Die Zahlenanordnung bleibt dieselbe. Man fügt einfach eine erklärende Kopfzeile und -spalte hinzu. So soll unser Beispiel aus Bild 27 eine Zusammenstellung der Einkäufe des Computerbenutzers Müller sein. Die Zeilen entsprechen den Monaten Januar, Februar, März und April (Bild 29).

	Disketten	Farbbänder	Druckerpapier
Januar	20	1	150
Februar	2	2	400
März	15	1	100
April	12	0	80

Bild 29. Die Matrix A als Tabelle. Müllers Einkäufe an Computerzubehör von Januar bis April

Die erste Spalte gibt die gekaufte Menge an Disketten, die zweite die an Farbbändern und die dritte die an Druckerpapier an (diese Beispiele wurden angeregt durch das Buch: Müller-Merbach, Operations Research, Berlin/ Frankfurt: Verlag Franz Vahlen 1969. Ein Buch, das kaufmännisch interessierten potentiellen Matrizennutzern sehr zu empfehlen ist). Müller hat noch einen Freund Meier, der ebenfalls einen Computer mit Zubehör sein eigen nennt und dessen Einkäufe uns zusammen mit denen von Müller im folgenden das Verständnis von Matrizen erleichtern sollen. Meiers Einkaufsmatrix finden Sie in Bild 30.

$$B=\begin{bmatrix} 10 & 1 & 500 \\ 30 & 1 & 200 \\ 5 & 2 & 100 \\ 15 & 1 & 300 \end{bmatrix}$$

Bild 30. Meiers Einkaufsmatrix

Sie sehen, daß Meier sehr produktiv ist. Das Element b(1,3) in seiner Einkaufsmatrix, nämlich 500 Blatt Computerpapier im Januar, deutet auf eine rege Korrespondenz hin. Aus der Bezeichnungsweise b(1,3) können Sie vielleicht schon ersehen, worauf wir nun hinaus wollen: Genauso werden ja in Basic die einzelnen Elemente von Arrays bezeichnet. Tatsächlich lassen sich Matrizen im Computer als zweidimensionale Arrays auffassen, ja im englischen Sprachgebrauch verschwimmen die Bedeutungen von »matrix« und »array«, sie sind in gewisser Weise fast Synonyme (Synonyme sind verschiedene Wörter für denselben Gegenstand). Nebenbei bemerkt: Genauso, wie wir eindimensionale Arrays definieren können (beispielsweise durch DIM A$(20)) gibt es auch Matrizen, wie die vergleichbare 1,20-Matrix oder die 20,1-Matrix. In diesen Fällen spricht man von »Vektoren«. Anders herum kann man auch Arrays mit mehr als zwei Dimensionen bilden. Ebenso gilt das auch für Matrizen. Wir beschränken uns aber im folgenden auf die zweidimensionalen Matrizen und Arrays.

Addieren von Matrizen

Aus unerfindlichen Gründen wollen Müller und Meier wissen, wie ihr gemeinsamer Verbrauch in den fraglichen Monaten war. Sie müssen daher ihre Einkaufsmatrizen zusammenzählen. Bild 31 zeigt Ihnen, wie das geschieht.

$$A+B=\begin{bmatrix} 20+10 & 1+1 & 150+500 \\ 2+30 & 2+1 & 400+200 \\ 15+5 & 1+2 & 100+100 \\ 12+15 & 0+1 & 80+300 \end{bmatrix}=\begin{bmatrix} 30 & 2 & 650 \\ 32 & 3 & 600 \\ 20 & 3 & 200 \\ 27 & 1 & 380 \end{bmatrix}=S$$

Bild 31. Addition der Einkaufsmatritzen Müller und Meier

Es werden einfach alle Elemente mit gleichen Indices addiert. Allgemein sehen Sie die Addition in Bild 32.

$$A+B=\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \\ a_{41}+b_{41} & a_{42}+b_{42} & a_{43}+b_{43} \end{bmatrix}=\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{bmatrix}=C$$

Bild 32. Addition zweier Matrizen allgemein ausgedrückt

Wichtig ist: Man kann nur Matrizen gleicher Zeilen- und Spaltenanzahl addieren. Für solche, die es genau wissen wollen: Es gilt bei der Matrizenaddition das kommutative Gesetz, die Summanden können vertauscht werden:
$A + B = B + A$

Außerdem gilt das Assoziativgesetz. Das bedeutet, daß auch bei mehr als drei Summanden die Reihenfolge beliebig ist:
$(A + B) + C = A + (B + C) = A + B + C$

Subtrahieren von Matrizen

Zum Abziehen zweier Matrizen voneinander braucht man eigentlich kaum Worte zu verlieren, denn das funktioniert genauso wie die Addition. Jeweils die Elemente mit gleichen Indices werden voneinander subtrahiert. Wollen unsere beiden Computerfreunde also wissen, wo ihre Verbrauchsdifferenzen liegen, dann bilden sie einfach A - B, wie in Bild 33.

$$A-B=\begin{bmatrix} 20-10 & 1-1 & 150-500 \\ 2-30 & 2-1 & 400-200 \\ 15-5 & 1-2 & 100-100 \\ 12-15 & 0-1 & 80-300 \end{bmatrix}=\begin{bmatrix} -10 & 0 & -350 \\ -28 & 1 & 200 \\ 10 & -1 & 0 \\ -3 & -1 & -220 \end{bmatrix}=D$$

Bild 33. Verbrauchsunterschiede von Müller und Meier

Meiers Verbrauch ist im Durchschnitt in diesen vier Monaten höher gewesen als Müllers, wie man an den vielen negativen Werten in der Differenz-Matrix D sehen kann.

Man kann Matrizen also beliebig voneinander abziehen oder addieren. Allerdings ist es wichtig, sich immer der Bedeutung der Zeilen und Spalten bewußt zu sein. Wenn in der zu Meier gehörenden Matrix B die Spalten anders angeordnet sind (beispielsweise in der Reihenfolge Disketten, Papier und Farbbänder) hat es wenig Sinn, die Summe

A + B zu bilden. Zuvor müssen beide Matrizen die gleiche Element-Anordnung aufweisen.

Multiplikation einer Matrix mit einem Faktor

Häufig kommt es vor, daß eine Matrix mit einer normalen Zahl malzunehmen ist. Will Meier bespielsweise wissen, wie hoch sein Verbrauch gewesen wäre, wenn er dreimal soviel Zeit zur Verfügung gehabt hätte, dann kann er das Produkt so bilden, wie in Bild 34 gezeigt wird.

$$3*B=\begin{bmatrix} 3*10 & 3*1 & 3*500 \\ 3*30 & 3*1 & 3*200 \\ 3*5 & 3*2 & 3*100 \\ 3*15 & 3*1 & 3*300 \end{bmatrix} = \begin{bmatrix} 30 & 3 & 1500 \\ 90 & 3 & 600 \\ 15 & 6 & 300 \\ 45 & 3 & 900 \end{bmatrix} =E$$

Bild 34. Meier berechnet den dreifachen Verbrauch

Jedes Element der Matrix wird mit dem Faktor multipliziert. Allgemein ausgedrückt, sehen Sie das in Bild 35.

$$K*A=\begin{bmatrix} K*a_{11} & K*a_{12} & K*a_{13} \\ K*a_{21} & K*a_{22} & K*a_{23} \\ K*a_{31} & K*a_{32} & K*a_{33} \end{bmatrix}$$

Bild 35. Multiplikation einer Matrix mit einem Faktor

Worin liegt denn nun der Vorteil dieser Matrizen? Herkömmliche Rechenverfahren schienen uns für die denkbaren Probleme ebenso einfach anwendbar zu sein. Vermutlich geht es Ihnen ähnlich. Das wird erst jetzt deutlicher, wo wir uns der Multiplikation von Matrizen untereinander zuwenden.

Matrizenmultiplikation

Benutzen wir hier wieder zum Erklären unser Beispiel mit Müller und Meier. In der Kleinstadt, in der unsere beiden Freunde leben, gibt es zwei Händler, die Computerzubehör führen: Vorteil und Reibach. Beide Händler sind bereit, monatliche Zahlung mit ihren Kunden zu vereinbaren, vorausgesetzt, daß man alle drei Warengruppen komplett bei jeweils nur einem von ihnen kauft. Müller und Meier müssen sich also entscheiden, bei welchem der beiden Händler sie im jeweils anstehenden Monat kaufen. Natürlich werden sie sich an den Preisen orientieren. Bild 36 zeigt eine Preistabelle beider Geschäfte.

	Vorteil	Reibach
Disketten	5,00	6,00
Farbbänder	38,00	35,00
Papier	0,08	0,07

Bild 36. Die Preistabelle der beiden Händler Vorteil und Reibach

Auch diese Tabelle kann man als Matrix, wir nennen sie die Preismatrix P, darstellen (siehe Bild 37).

$$P=\begin{bmatrix} 5 & 6 \\ 38 & 35 \\ 0{,}08 & 0{,}07 \end{bmatrix}$$

Bild 37. Die Preismatrix P

Wenn Meier ein vorausplanender Mensch ist, wird er nun vorher schon überlegen, in welchem Monat er bei Vorteil und in welchem bei Reibach kauft. Im Monat Januar ergäben sich für ihn bei Vorteil folgende Kosten:
20 Disketten mal 5 Mark plus
1 Farbband zu 38 Mark plus
150 Blatt Papier zu je 0,08 Mark
150,00 Mark wäre die Summe davon. Bei Händler Reibach ergäbe sich:
20*6,00 + 1*35,00 + 150*0,07 = 165,50 Mark

Auf die gleiche Weise berechnet er nun die Kosten bei beiden Geschäften in den anderen Monaten und erhält folgende Tabelle (siehe Bild 38).

	bei Vorteil	bei Reibach
Januar	150,00	165,50
Februar	118,00	110,00
März	121,00	132,00
April	66,40	77,60

Bild 38. Müllers Preisübersicht

Außer im Februar sollte Müller also bei Vorteil kaufen. Mit den Beispielrechnungen haben wir schon das Prinzip der Matrizenmultiplikation nachvollzogen: Die Zahlen einer Zeile aus der Matrix A wurden mit den Zahlen einer Spalte der Matrix P multipliziert und daraus die Summe gebildet. Auf diese Weise erhalten wir ein Element der Ergebnismatrix, die wir in Bild 38 als Tabelle dargestellt haben. So ergibt die Verknüpfung der ersten Zeile von A mit der zweiten Spalte von P das Glied m(1,2) der Ergebnismatrix M. Falk hat 1951 ein Schema vorgestellt, das für das Verständnis dieser Multiplikation vorteilhaft ist. Beide Matrizen und die Ergebnismatrix packt man in ein Schema wie in Bild 39.

				P		
	A*P=M			o	o	o
				o	o	o
				o	o	o
				o	o	o
A				M		
o	o	o	o	o	o	o
o	o	o	o	o	o	o
o	o	o	o	o	o	o
o	o	o	o	o	o	o

Bild 39. Schema zur Matrizen-Multiplikation nach Falk

Auf unser Beispiel mit Herrn Müller angewendet, ergibt sich damit (ausführlich beschrieben zum Nachvollziehen) das Schema in Bild 40.

Bild 41 zeigt Ihnen die Matrizenmultiplikation nach Falk in allgemeiner Schreibweise.

Besonders dann, wenn Sie sich in Bild 41 einmal die Bildung der verschiedenen m-Elemente ansehen, fällt Ihnen

	5 38 0,08	6 35 0,07
20 1 150	20*5+1*38+150*0,08	20*6+1*35+150*0,07
2 2 400	2*5+2*38+400*0,08	2*6+2*35+400*0,07
15 1 100	15*5+1*38+100*0,08	15*6+1*35+100*0,07
12 0 80	12*5+0*38+ 80*0,08	12*6+0*35+ 80*0,07

Bild 40. Müllers Kostenrechnung mit dem Falk-Schema

	p_{11} p_{21} p_{31}	p_{12} p_{22} p_{32}
a_{11} a_{12} a_{13}	$m_{11}=a_{11}*p_{11}+a_{12}*p_{21}+a_{13}*p_{31}$	$m_{12}=a_{11}*p_{12}+a_{12}*p_{22}+a_{13}*p_{32}$
a_{21} a_{22} a_{23}	$m_{21}=a_{21}*p_{11}+a_{22}*p_{21}+a_{23}*P_{31}$	$m_{22}=a_{21}*p_{12}+a_{22}*p_{22}+a_{23}*p_{32}$
a_{31} a_{32} a_{33}	$m_{31}=a_{31}*p_{11}+a_{32}*p_{21}+a_{33}*p_{31}$	$m_{32}=a_{31}*p_{12}+a_{32}*p_{22}+a_{33}*p_{32}$

Bild 41. Allgemeine Schreibweise einer Matrizen-multiplikation nach Falk

sicherlich auf, wie sich die Indices der Faktoren a und p mit schöner Regelmäßigkeit verändern. Erinnern Sie sich außerdem daran, daß wir ein Element aus einer Matrix auch als Array-Element verstehen können, dann erkennen Sie sicher schnell, wie einfach derartige Rechnungen per Computer durchzuführen sind. Dazu kommen wir gleich noch.

Damit Sie lernen, ein wenig mit diesem Prinzip zu arbeiten, bitten wir Sie, dasselbe mal an der Kostenrechnung von Herrn Meier durchzuführen.

Sie haben also folgende Rechnung zu vollziehen: N = B * P (N sei die Ergebnismatrix von Meier, B ist seine Planung (siehe Bild 30), P ist die Preismatrix aus Bild 37).

Wenn wir alle richtig gerechnet haben, dann sollten Sie als Ergebnis die Matrix aus Bild 42 erhalten.

$$N=\begin{bmatrix} 128 & 130 \\ 204 & 229 \\ 109 & 107 \\ 137 & 146 \end{bmatrix}$$

Bild 42. Meiers Ergebnismatrix N

Lediglich im März sollte Meier bei Reibach kaufen, ansonsten ist der Kauf bei Vorteil für ihn von Vorteil.

Bei all dieser Rechnerei werden Ihnen folgende Eigenarten aufgefallen sein (wir gehen mal von N = B * P aus):
- N hat genauso viele Zeilen wie B und ebenso viele Spalten wie P.
- Eine Multiplikation ist nur möglich, wenn die Anzahl der Spalten von B gleich der Anzahl der Zeilen von P ist. Das sollte einleuchten: Für jede Warenanzahl in B (beispielsweise zehn Disketten) muß auch ein Preis angegeben sein, wenn man die Rechnung überhaupt durchführen will.
- Bisher nicht auffallen - weil wir mit Matrizen gearbeitet haben, bei denen eine Vertauschung zur Multiplikation nicht möglich war - konnte Ihnen die Tatsache, daß man die Reihenfolge der Matrizen bei der Multiplikation nicht verändern darf. Nehmen wir an, wir hätten Matrizen vorliegen, bei denen man die Faktoren vertauschen kann, dann gilt hier - im Gegensatz zur normalen Multiplikation -, daß A * B nicht gleich B * A ist! Nehmen wir mal - zur Übung - zwei einfache Matrizen (siehe Bild 43).

$$A=\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 3 \\ 3 & 0 & 1 \end{bmatrix} \qquad B=\begin{bmatrix} 0 & 0 & 1 \\ 3 & 2 & 1 \\ 2 & 0 & 3 \end{bmatrix}$$

Bild 43. Zwei Matrizen zur eigenen Übung

Versuchen Sie nun, beide Multiplikationen mittels des Falk-Schemas durchzuführen, also zu rechnen A * B und auch B * A. Ihre Ergebnisse sollten denen aus Bild 44 entsprechen.

$$A*B=\begin{bmatrix} 6 & 4 & 3 \\ 9 & 2 & 12 \\ 2 & 0 & 6 \end{bmatrix} \qquad B*A=\begin{bmatrix} 3 & 0 & 1 \\ 10 & 8 & 7 \\ 11 & 4 & 3 \end{bmatrix}$$

Bild 44. Die Ergebnismatrizen der Übung

Sie sehen, es entstehen unterschiedliche Ergebnismatrizen. Das heißt, bei der Multiplikation nach dem Falk-Schema ist A*B nicht identisch mit B*A. Damit sei es genug mit der Einführung in die Matrizenrechnung. Sehen wir uns nun an, wie wir diese Gebilde in der Computergrafik nutzen können.

Ein Punkt als Matrix

Wir hatten festgestellt, daß eine Matrix einfach eine geordnete rechteckige Darstellung von Elementen ist, beispielsweise von Zahlen. Nichts hindert uns also daran, auch die Koordinaten eines Punktes (siehe Bild 45) als eine Matrix aufzufassen.

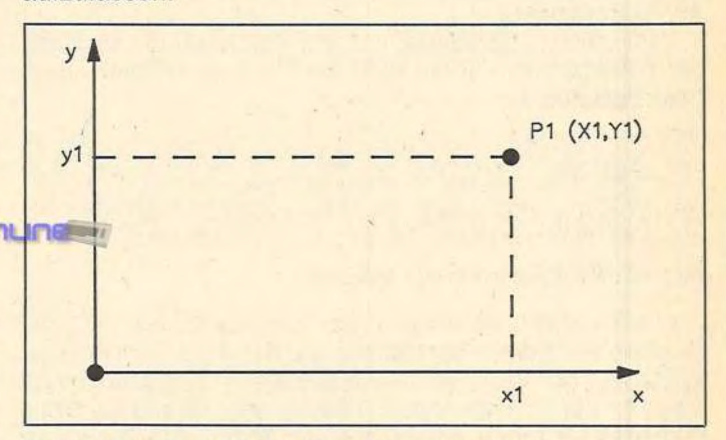


Bild 45. Der Punkt P1 im Koordinatensystem

Einem Punkt P1(X1,Y1) entspricht so also eine 1,2-Matrix, nämlich
(X1 Y1)

Nun können wir ausprobieren, was geschieht, wenn wir diese kleine Matrix allerlei Rechnereien aussetzen.

Was passiert, wenn wir die 1,2-Matrix eines Punktes mit einer 2,2-Matrix malnehmen? Bild 46 zeigt Ihnen ein Beispiel:

P1 = (X1 Y1) $\qquad E=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

P1*E=(X1 Y1)

P1*E	1 0 0 1
X1 Y1	X1 Y1

Bild 46. Multiplikation von P1 mit der Matrix E

P1 ist unser Punkt, E die 2,2-Matrix. Das Ergebnis ist wieder eine 1,2-Matrix. Durch die Multiplikation ist P1 gar nicht verändert worden! Wir haben hier eine besondere Matrix gewählt, nämlich die sogenannte Einheitsmatrix, die auch als E bezeichnet wird.

Eine Einheitsmatrix wie in Bild 46 erkennt man immer an zwei Kriterien: Es ist eine quadratische Matrix, also eine, bei der die Anzahl der Zeilen und Spalten gleich ist, und sie enthält auf der Diagonalen von links oben nach rechts unten nur Einsen, während alle anderen Elemente gleich Null sind.

$$P1 = (X1\ Y1) \qquad T1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$P2 = P1 * T1 = (2 * X1\ Y1)$$

	2 0
	0 1
X1 Y1	2* X1Y1

Bild 47. P1 wird mit der Matrix T1 multipliziert

Sehen wir uns mal eine andere 2,2-Matrix an, die wir T1 nennen. Bild 47 zeigt Ihnen, was hier die Multiplikation erzeugt.

P2 – der Punkt, der sich in Bild 47 aus der Multiplikation P1 * T1 ergibt – hat doppelt so große X-Koordinaten als P1. Setzen wir statt der 2 in T1 nun 0.5 ein, dann ergibt sich eine Ergebnismatrix mit halbierten X-Werten. Probieren Sie es doch einfach einmal aus!

Einfache Transformation

Die 2,2-Matrix T1 ist das, was wir in den letzten Abschnitten angekündigt haben: eine Transformationsmatrix. Eine sehr einfache zwar, aber immerhin wissen Sie nun, wie man durch Multiplikation einer Punktmatrix mit einer Transformationsmatrix eine Transformation ausführen kann. Ein und denselben Punkt P1 können wir nun als P2 in Koordinatensystemen mit gestreckter oder auch gestauchter X-Achse darstellen.

Vermutlich ahnen Sie schon, wie man dasselbe auch mit der Y-Achse durchführen kann. Bild 48 zeigt es Ihnen an einem Beispiel:

$$P1 = (X1\ Y1) \qquad T2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$P2 = P1 * T2 = (X1\ 2 * Y1)$$

	1 0
	0 2
X1 Y1	X1 2*Y1

Bild 48. Die Skalierung der Y-Achse

Wie Sie im Bild 48 sehen, ergibt sich aus P2 = P1 * T2 die Verdoppelung der Y-Koordinate, wenn das Element in Zeile 2, Spalte 2 der Transformationsmatrix eine 2 ist. Man nennt diese Art der Transformation, die wir nun mit X und auch mit Y ausgeführt haben, eine Skalierung. Mittels Skalierung ist schon eine große Hürde genommen, die auf dem Weg von Weltkoordinaten zu Bildschirmkoordinaten steht.

Die Skalierung

Jetzt aber wollen wir noch klären, was zu tun ist, wenn sowohl in X-, als auch in Y-Richtung skaliert werden muß. Das wird ohnehin meistens der Fall sein. T1 hatten wir die Transformationsmatrix genannt, die in X-Richtung skalierte, T2 war die für die Y-Achse. Wir können beide Rechnungen nacheinander ausführen. Dann steht für die X-Skalierung:

P2 = P1 * T1

und für die Y-Richtung anschließend:

P3 = P2 * T2

Setzen wir den Ausdruck für P2 in die zweite Gleichung ein, dann ergibt sich:

P3 = P1 * T1 * T2

Im letzten Abschnitt hatten wir zwar festgestellt, daß man die Reihenfolge der Faktoren bei einer Matrizenmultiplikation nicht vertauschen darf. Andererseits aber gilt das sogenannte Assoziativgesetz, man kann also folgendes tun:

A * B * C = A * (B * C) = (A * B) * C

Es spielt keine Rolle, ob man zuerst P1 mit T1 malnimmt und dieses Ergebnis dann mal T2, oder ob man zuerst T1 mal T2 rechnet und das Ergebnis dann als neuen Faktor bei der Multiplikation mit P1 verwendet. Das erlaubt uns die Berechnung einer allgemeinen Skalierungsmatrix, in der sowohl die X- als auch die Y-Richtung gleichzeitig erfaßt werden. Bild 49 zeigt uns an unseren obigen Beispielen, wie solch eine Skalierungsmatrix durch Bilden von T1 * T2 entsteht:

$$T1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad T2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$S = T1 * T2 \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

	1 0
	0 2
2 0	2 0
0 1	0 2

Bild 49. Die Bildung einer gemeinsamen Skalierungsmatrix für die X- und die Y-Achse

Halten wir nun fest, wie eine allgemeine Skalierungsmatrix aussehen muß, die wir für jeden vorkommenden Fall anwenden können (Bild 50).

$$S = \begin{bmatrix} SX & 0 \\ 0 & SY \end{bmatrix}$$

Bild 50. Die allgemeine Skalierungsmatrix S

Dabei ist SX der Skalierungsfaktor der X-Achse und SY der in Y-Richtung. Damit sollen die Skalierungen erst einmal beiseite gelegt werden. Behalten Sie die Matrix S aber in Ihrem Gedächtnis. Wir werden sie später wieder hervorholen.

Die Rotation

Außer der Skalierung sind es noch zwei Veränderungen von grafischen Abbildungen, die uns interessieren werden: die Rotation und die Translation. Unter Translation versteht man einfach die Verschiebung eines Koordinatensystems, unter Rotation die Drehung. Zur Translation kommen wir später. Beginnen wir zunächst einmal mit der Rotation.

Bild 51 zeigt die Verhältnisse für den Fall einer Rotation um den Ursprungspunkt (also den Punkt 0,0).

Ein Punkt P1 mit den Koordinaten X1 und Y1 ist durch Rotation um den Winkel w2 überführt worden in einen Punkt P2 (Koordinaten X2 und Y2). Uns interessiert, wie man die neuen Koordinaten X2 und Y2 aus den alten berechnen kann. Die Entfernung L des Punktes vom Koordinatenur-

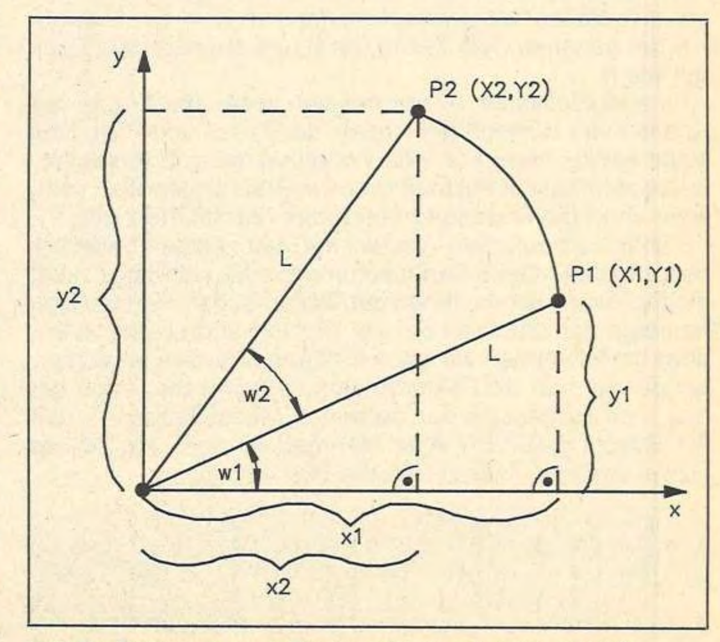


Bild 51. Rotieren eines Punktes P1 um den Winkel W2

sprung 0 ist konstant geblieben. Vor der Rotation kann man für Sinus und Cosinus schreiben:
SIN(w1) = Y1/L und COS(w1) = X1/L
Daraus folgt durch Umstellen der Gleichungen:
Y1 = L*SIN(w1) und X1 = L*COS(w1)
Das behalten wir erst einmal im Gedächtnis und sehen uns nun die Verhältnisse nach der Rotation an. Aus w1 ist nun die Winkelsumme w1+w2 geworden. Für den Sinus und den Cosinus kann man nun schreiben:
SIN(w1+w2) = Y2/L und COS(w1+w2) = X2/L
Nach dem Umstellen ergibt sich hier ähnlich wie vorhin:
Y2 = L*SIN(w1+w2) und X2 = L*COS(w1+w2)
Kluge Mathematiker haben sich den Kopf zerbrochen und schließlich festgestellt, daß man für den Sinus einer Winkelsumme auch schreiben kann:
SIN(w1+w2) = COS(w2)*SIN(w1) + SIN(w2)*COS(w1)
Analog dazu fanden sie für den Cosinus:
COS(w1+w2) = COS(w2)*COS(w1) - SIN(w2)*SIN(w1)
Diese beiden Beziehungen setzen wir in die von uns gefundenen für Y2 und X2 ein und erhalten auf diese Weise:
Y2 = L*(COS(w2)*SIN(w1) + SIN(w2)*COS(w1)) und
X2 = L*(COS(w2)*COS(w1) - SIN(w2)*SIN(w1))
Das Auflösen der äußeren Klammer liefert uns:
Y2 = COS(w2)*SIN(w1)*L + SIN(w2)*COS(w1)*L und
X2 = COS(w2)*COS(w1)*L - SIN(w2)*SIN(w1)*L
Nun holen wir uns wieder die vorhin im Gedächtnis behaltenen Gleichungen für Y1 und X1 hervor. Wenn Sie genau hinsehen, dann stellen Sie fest, daß wir jeweils zweimal pro Gleichung diese Beziehungen enthalten finden (beispielsweise in der Gleichung für Y2 am Ende des ersten Summanden steht SIN(w1)*L, was ja Y1 entspricht). Wir verwenden also diese Einsetzmethode und gelangen so zu:
Y2 = COS(w2)*Y1 + SIN(w2)*X1 und
X2 = COS(w2)*X1 - SIN(w2)*Y1
Damit haben wir nun zwei Gleichungen, die es uns erlauben, für jeden Punkt P1 mit den Koordinaten X1 und Y1 nach einer Drehung um den Winkel w2 die neuen Koordinaten X2 und Y2 zu berechnen. Nun können Sie es mit den Kenntnissen über die Matrizenmultiplikation nachprüfen: Dasselbe Ergebnis folgt nämlich auch aus der Multiplikation der Punktmatrix (X1 Y1) mit einer Transformationsmatrix R (also P1*R), was Ihnen Bild 52 zeigt.

$$(X2\ Y2) = (X1\ Y1) * \begin{bmatrix} \cos(W2) & \sin(W2) \\ -\sin(W2) & \cos(W2) \end{bmatrix}$$

Bild 52. Diese Transformation führt zur Berechnung der Punktkoordinaten nach einer Rotation

R ist die sogenannte Rotationsmatrix, die Sie sich gut merken sollten (Bild 53).
R in Bild 53 ist die allgemeine Rotationsmatrix, die in dieser Form für jeden Winkel w2 angewendet werden kann. w2

$$R = \begin{bmatrix} \cos(W2) & \sin(W2) \\ -\sin(W2) & \cos(W2) \end{bmatrix}$$

Bild 53. Die Rotationsmatrix R

wird positiv gerechnet bei Drehungen gegen den Uhrzeigersinn und bei Drehungen mit dem Uhrzeigersinn negativ. Handelt es sich also um solch eine negative Rotation, dann erhält man für R einen Ausdruck wie in Bild 54.

$$R = \begin{bmatrix} \cos(-W2) & \sin(-W2) \\ -\sin(-W2) & \cos(-W2) \end{bmatrix}$$

Bild 54. Die Rotationsmatrix bei einer Drehung im Uhrzeigersinn

Man kann sich in einem solchen Fall wieder einige mathematische Erkenntnisse zunutze machen, die die Winkelfunktionen negativer Winkel betreffen. Es gilt nämlich:
COS(-w2) = COS(w2) und
SIN (-w2) = -SIN(w2)
Damit vermeiden wir negative Winkel, und die veränderte Rotationsmatrix für Drehungen mit dem Uhrzeigersinn zeigt Ihnen dann Bild 55.

$$R = \begin{bmatrix} \cos(W2) & -\sin(W2) \\ \sin(W2) & \cos(W2) \end{bmatrix}$$

Bild 55. Die veränderte Rotationsmatrix für Drehungen im Uhrzeigersinn

Jetzt kennen wir die Matrizen zur Skalierung und zur Rotation. Die dritte Manipulation von Punkten wäre die Verschiebung oder Translation, der wir uns nun zuwenden.

Die Translation

Damit Sie sich die Unterschiede aller drei Operationen, die wir behandeln, auch vorstellen können, sind sie in Bild 56 (Seite 146) einmal aufgeführt.
Um eine Verschiebung in der Fläche durch Transformationen ausdrücken zu können, braucht man anstelle der bislang verwendeten 2,2-Matrizen eine 3,3-Matrix. Weil wir später – und da liegt überhaupt der Vorzug dieser Methode – auch Kombinationen dieser Transformationen durchführen werden, ist es sinnvoll, auch die bisher gefundenen Matrizen S und R als 3,3-Matrix zu schreiben (beispielsweise kann man dann eine Drehung, eine Skalierung und eine Translation mittels einer einzigen Matrix ausführen).

Spezialfälle der Transformation

Ebenso werden die Punkte nun durch drei Angaben statt wie bisher nur durch zwei (nämlich X1,Y1) beschrieben. Aber der Reihe nach:
Für die Punkte erfindet man noch einen Dummywert (also einen, der lediglich für die Bequemlichkeit der Rechnung geschaffen wird, der aber keine konkrete Bedeutung hat). Den nennen wir w. Unser Punkt
P1 : (X1 Y1)
heißt nun:
P1 : (X1*w Y1*w w)
Praktischerweise hat w einfach den Wert 1. Sie werden sich vielleicht erinnern, daß man bei der Matrizenmultiplikation auf die Anzahl der Zeilen und Spalten achten mußte: Das allein ist der Grund, der dieses w erforderlich macht. Wie sehen nun S und R als 3,3-Matrizen aus?
Man kann eine Matrix erweitern, ohne ihren Wert zu verändern, indem man eine Zeile und eine Spalte hinzufügt, die nur Nullen enthalten. In der Diagonalen kommt dann noch eine 1 dazu. Bild 57 zeigt Ihnen das an unserer Skalierungsmatrix.
Bild 58 soll Ihnen die Vorgehensweise zeigen, mit der nun gerechnet wird.
Das korrekte Ergebnis erhält man einfach durch Weglassen von w. Auch die Rotationsmatrix wird erweitert und lautet nun so, wie es in Bild 59 gezeigt wird.
Die Vorgehensweise beim Rechnen erfolgt genauso, wie wir es schon bei der erweiterten Skalierungsmatrix gesehen hatten. Damit sind die Anpassungen erledigt. Sehen wir uns nun die Ursache für diese Veränderungen an: die Translationsmatrix.

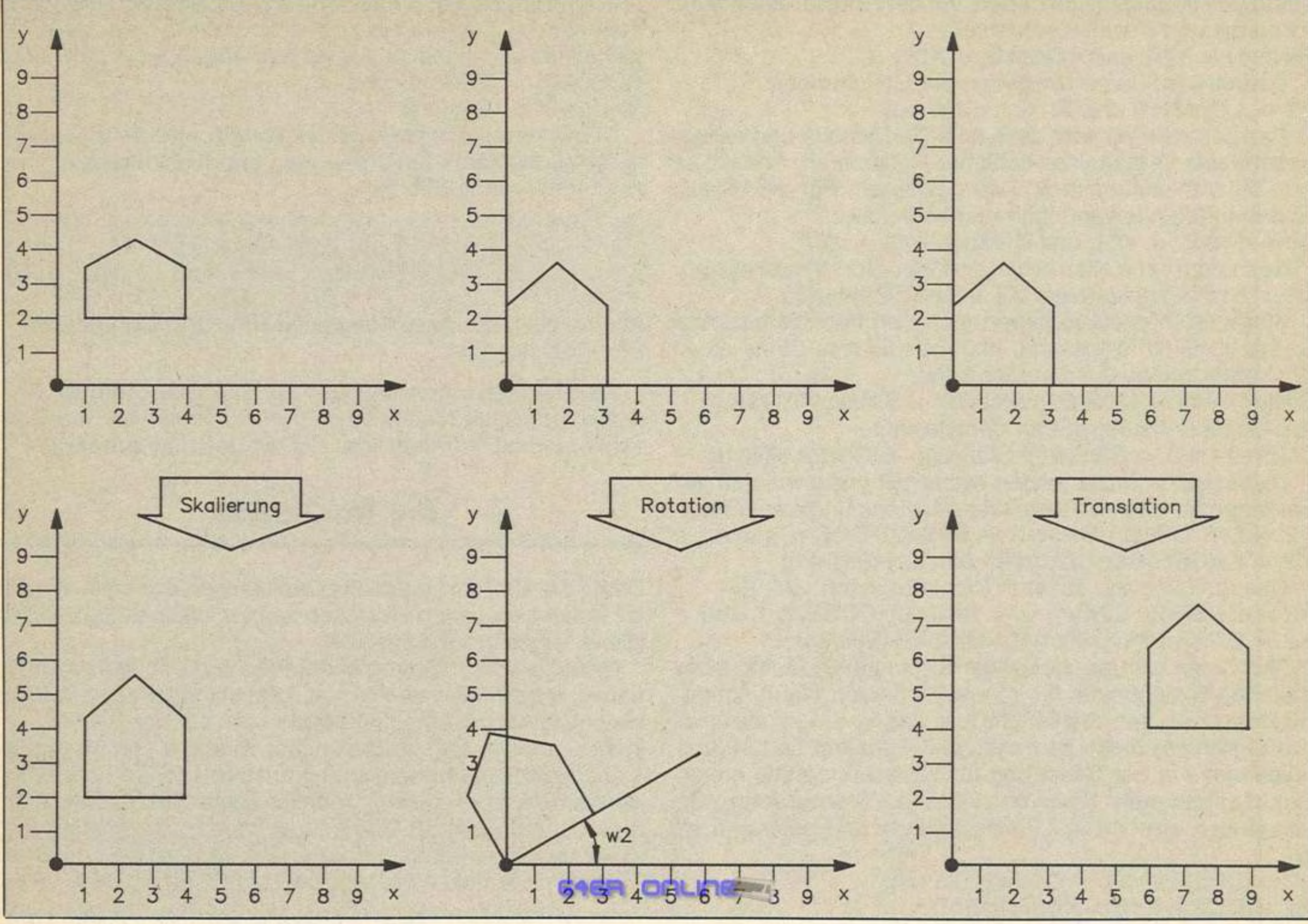


Bild 56. Die drei Arten der Transformation

$$S=\begin{bmatrix} SX & 0 & 0 \\ 0 & SY & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Bild 57. Die Skalierungsmatrix als 3,3-Matrix

P1 = (X1*W Y1*W W)

$$S=\begin{bmatrix} SX & 0 & 0 \\ 0 & SY & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

	SX	0	0
	0	SY	0
	0	0	1
X1*W Y1*W W	SX*X1*W	SY*Y1*W	W

P2 = P1*S = (SX*X1*W SY*Y1*W W) = (SX*X1 SY*Y1)

Bild 58. So wird mit der neuen Skalierungsmatrix gerechnet

$$R=\begin{bmatrix} \cos(W2) & \sin(W2) & 0 \\ -\sin(W2) & \cos(W2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Bild 59. Die neue 3,3-Rotationsmatrix

In Bild 60 finden Sie die allgemeine Form dieser Matrix. TX ist die Verschiebung in X-, TY die in die Y-Richtung.

$$T=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ TX & TY & 1 \end{bmatrix}$$

Bild 60. Die allgemeine Translationsmatrix T

Die Anwendung auf einen Punkt P1 (X1,Y1) als Operation
P2 = P1 * T
zeigt Ihnen das Bild 61.

P1 = (X1*W Y1*W W)

	1	0	0
	0	1	0
	TX	TY	1
X1*W Y1*W W	X1*W+TX*W	Y1*W+TY*W	W

$$T=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ TX & TY & 1 \end{bmatrix}$$

P2 = P1*T = (W[X1+TX] W[Y1+TX] W) = (X1+TX Y1+TY)

Bild 61. Anwendung der Translationsmatrix T auf einen Punkt P1

Die normalen Koordinaten ergeben sich wieder durch einfaches Weglassen von w.

Wenn Sie sich das Ergebnis genau ansehen, werden Sie sicher fragen, weshalb das alles auf so komplizierte Weise

erfolgen muß. Daß man TX zur X-Koordinate und TY zur Y-Koordinate zu addieren hat, sei vorher auch schon klar gewesen. Damit haben Sie natürlich recht. Jede einzelne dieser Matrizen, S, R oder T für sich allein genommen (R allerdings davon ausgenommen), kann man eigentlich ebensogut auch ersetzen durch normale Rechenvorgänge, die keinerlei Matrizen erfordern. Der Vorteil dieser Art der Transformation zeigt sich erst, wenn man bedenkt, daß sie sehr computergerecht abläuft. Man kann im Prinzip ein und dasselbe Programm für alle Operationen verwenden. Lediglich die Elemente der Matrix verändern sich.

In noch viel stärkerem Maß aber erfahren wir die Vorteile, wenn mehrere Transformationen zusammenfallen. Auch dann genügt immer noch eine einzige Matrix. Das gleiche Programm kann auch hier verwendet werden. Wieder sind nur die Elemente der Matrix andere. Schließlich sollten Sie noch bedenken, daß wir uns immer noch mit den einfachsten grafischen Objekten (nämlich den Punkten) im einfachsten Koordinatensystem (nämlich dem ebenen kartesischen System) befassen. Wenn wir später in die höheren Dimensionen aufsteigen, werden die Rechnungen ohne Matrizen manchmal reichlich verwirrend.

Die kombinierten 2D-Transformationen

Jetzt sind wir soweit, daß wir 2D-Grafik betreiben können. Nehmen wir ein beliebiges Objekt, beispielsweise ein Haus, das direkt an seinem Bildschirmplatz gedreht und verkleinert werden soll. Es sind also eine Reihe von Transformationen hintereinander auszuführen, die wir in eine einzige Transformationsmatrix zusammenfassen werden. Bild 62 zeigt Ihnen den Weg in Einzelschritten.

Was hat der Programmierer zu tun?

a) Er wählt ein den Maßen des Hauses angepaßtes Weltkoordinatensystem und legt darin die charakteristischen Punkte fest.

Die Reihenfolge ist wichtig

Nehmen wir an, unser Haus hätte eine Breite von 10m, eine Höhe bis zum Giebel von 13m und bis zur Dachtraufe von 7m. Die Gestalt wäre die in Bild 62 gezeigte. Damit genügend Platz für alle Betrachtungen ist, wählen wir ein Weltkoordinatensystem, das von 0m bis 30m horizontal und von 0m bis 25m vertikal reicht. Der linke untere Eckpunkt des Hauses befände sich an der Stelle P1(10,5). Dann gilt für die anderen Eckpunkte (im Uhrzeigersinn aufgezählt): P2(10, 12), P3(15,18), P4(20,12) und P5(20,5). Das Zeichnen des Hauses ist dann einfach damit erledigt, daß man diese Punkte in ihrer Reihenfolge durch Linien verknüpft und zu guter Letzt noch P5 mit P1 verbindet. Wir haben damit die Ausgangssituation in Bild 62 hergestellt.

b) Zerlegen einer komplexen Veränderung des Objektes in eine Reihe von Translationen, Rotationen und Skalierungen.

In Bild 62 sehen Sie diese Reihenfolge. Wenn wir wieder den linken unteren Punkt P1 des Hauses als Bezugspunkt ansehen, um den die Drehung und an dem die Skalierung stattfinden soll, dann beginnt unsere Arbeit mit einer Verschiebung des Objektes an den Ursprung. Würde man das nicht tun, sondern gleich drehen, dann bekäme man ein völlig anderes Ergebnis, das Sie in Bild 63 erkennen können.

Jede Rotation und jede Skalierung nimmt nämlich immer den Ursprung als Bezugspunkt an. Bei Rotationen um ei-

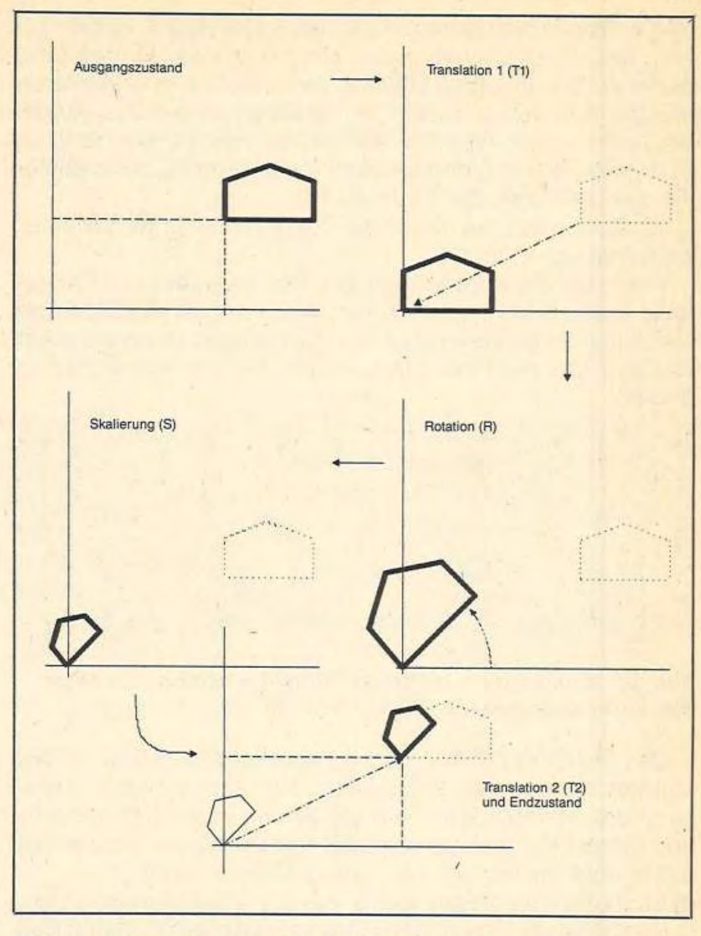


Bild 62. Hier sehen Sie eine kombinierte Transformation in der Reihenfolge T1, S, T2

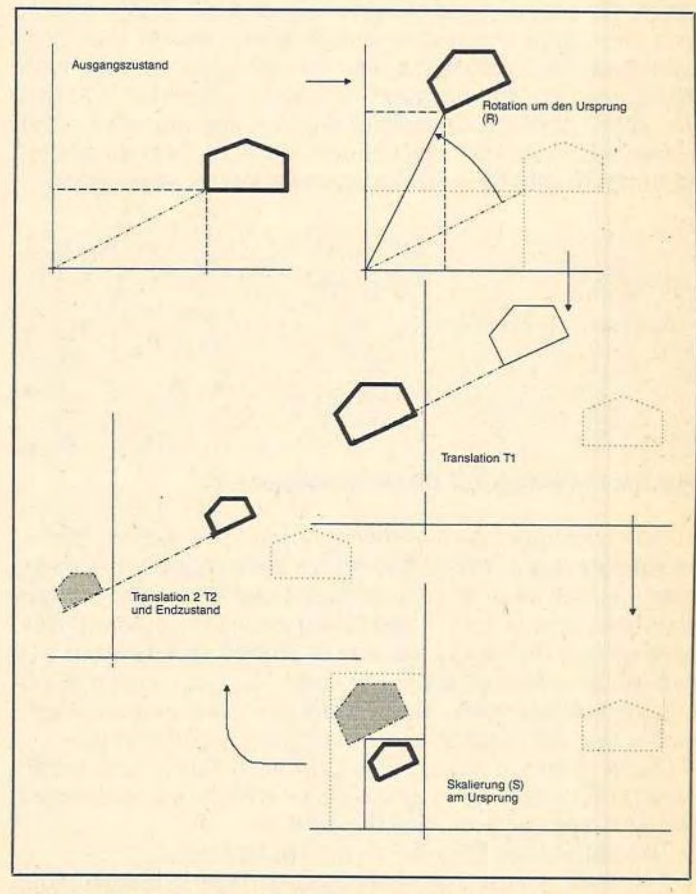


Bild 63. Nun eine kombinierte Transformation in der Reihenfolge R, T1, S, T2

nen anderen Bezugspunkt gilt daher die Regel: Immer zuerst den Bezugspunkt in den Ursprung verschieben (und damit auch das ganze Objekt), dann drehen und skalieren und danach wieder zurück verschieben an den Ausgangsort. Nennen wir die erste Translation also T1, die Rotation R, die Skalierung S und die Rücktranslation T2, dann gilt für die geplante Abfolge T1, R, S, T2.

c) Multiplikation der Matrizen der Einzelschritte zu einer kombinierten Matrix.

Wenn wir die Koordinaten des Bezugspunktes P1 allgemein als xc und yc bezeichnen, W als den Drehwinkel und die Skalierungsfaktoren Sx und Sy nennen, dann erfordert das von uns die Multiplikation der vier Einzelmatrizen in Bild 64.

$$\begin{bmatrix}1&0&0\\0&1&0\\-XC&-YC&1\end{bmatrix} * \begin{bmatrix}cosW&sinW&0\\-sinW&cosW&0\\0&0&1\end{bmatrix} * \begin{bmatrix}Sx&0&0\\0&Sy&0\\0&0&1\end{bmatrix} * \begin{bmatrix}1&0&0\\0&1&0\\XC&YC&1\end{bmatrix}$$

$$= \begin{bmatrix}Sx*cosW & Sx*sinW & 0\\ -Sy*sinW & Sy*cosW & 0\\ xc*(1-Sx*cosW)+Sy*yc*sinW & yc*(1-Sy*cosW)-Sx*xc*sinW & 1\end{bmatrix}$$

Bild 64. Kombination mehrerer Transformationen zu einer Transformationsmatrix: M=T1*R*S*T2

Das Ergebnis M finden Sie dann ebenfalls im Bild 64. Sie können es nachvollziehen, wenn Sie je zwei Matrizen miteinander malnehmen – wie wir es vorhin gelernt haben – und diese Produkte dann wieder multiplizieren. Achten Sie dabei aber immer auf die richtige Reihenfolge!

d) Multiplikation dieser Matrix mit der Anpassungsmatrix.

Das ist etwas Neues! Die Anpassungsmatrix bewirkt den Übergang vom Weltkoordinatensystem auf die Bildschirmkoordinaten. Alles bisher Unternommene hat sich immer noch im Weltsystem abgespielt. Die Anpassungsmatrix nimmt die nötigen Skalierungen in X- und in Y-Richtung vor, und sie macht aus dem linkshändigen System des Bildschirmes ein rechtshändiges. Das geschieht durch eine 3D-Operation, nämlich durch eine Rotation um die X-Achse um 180 Grad. Wir haben diese Matrix daher hier noch nicht entwickelt. Damit Sie aber diesen Teil auch kennenlernen, ist Ihnen in Bild 65 eine Anpassungsmatrix angegeben.

$$\text{Anpassungsmatrix} = \begin{bmatrix}\frac{XM}{XO-XU} & 0 & 0\\ 0 & -\frac{YM}{YO-YU} & 0\\ -\frac{XM*XU}{XO-XU} & \frac{YM*YO}{YO-YU} & 1\end{bmatrix}$$

Bild 65. Die Matrix zur Bildschirmanpassung

Hier bedeutet XM die maximal mögliche X-Bildschirmkoordinate (bei uns also 319), YM dasselbe für die Y-Koordinate (bei uns also 199). Der kleinste und der größte X-Wert des Weltsystems wird XU und XO genannt (in unserem Beispiel ist das 0 und 30), bei den Y-Werten ist das dann YU und YO (in unserem Beispiel 0 und 25). Die Multiplikation unserer bisher entwickelten Matrix mit dieser Anpassungsmatrix und das Ergebnis davon finden Sie in Bild 66.

Diese nun gefundene Gesamtmatrix verwandelt jeden unserer Hauspunkte in einen des gedrehten und skalierten Hauses, angepaßt an den Bildschirm.

e) Berechnen der Elemente der Gesamtmatrix.

Es ist jetzt an der Zeit, im Programm oder außerhalb, die einzelnen Elemente der Gesamtmatrix zu berechnen. Aus Bild 67 folgt für unsere Operationen nun:

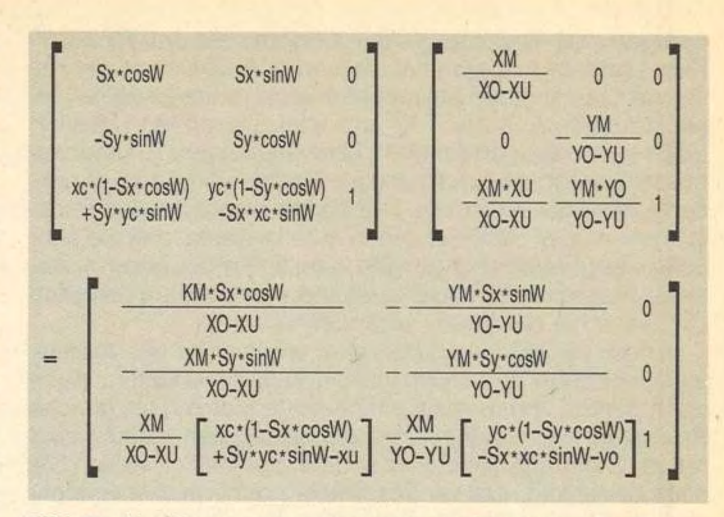


Bild 66. Kombination der allgemeinen Transformationsmatrix T mit der Matrix zur Bildschirmanpassung (M): A=T*M

r11 = XM*Sx*cos(W)/(XO-XU)
r12 = -YM*Sx*sin(W)/(YO-YU)
r21 = -XM*Sy*sin(W)/(XO-XU)
r22 = -YM*Sy*cos(W)/(YO-YU)
t1 = XM*(Xc*(1-Sx*cos(W)) + Sy*Yc*sin(W)-XU)/(XO-XU)
t2 = -YM*(Yc*(1-Sy*cos(W))-Sx*Xc*sin(W)-YO)/(YO-YU)

Wenn im Rahmen eines Programms alle Parameter bekannt sind, ist nun auch die Gesamtmatrix voll berechenbar.

f) Aufstellen der beiden Transformationsgleichungen.

Auch das ist Ihnen neu. Unsere Arbeit mit den Matrizen ist nämlich jetzt beendet. Was jetzt folgen würde, wäre die Multiplikation aller Punkte (als n,3-Matrix, wobei n die Anzahl der bildwichtigen Punkte ist, auf unser Haus bezogen also n=5) mit der jetzt bekannten Gesamtmatrix, also eine einfache Matrizenmultiplikation. Wir brauchen sie aber tatsächlich gar nicht mehr im Programm ausführen lassen. Den Grund dafür zeigt Ihnen Bild 67.

Dort ist ein Punkt (X,Y) mit der vereinfachten Gesamtmatrix malgenommen worden (W ist hier der schon bekannte Dummy-Wert). Als Ergebnis finden sich einfach zwei Gleichungen:

X' = r11 * X + r21 * Y + t1
Y' = r12 * X + r22 * Y + t2

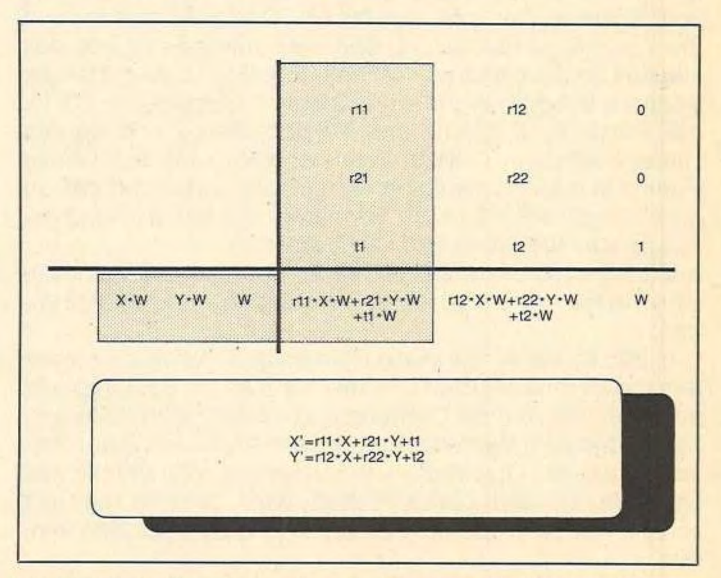


Bild 67. Jede Transformationsmatrix kann im Endeffekt zu zwei Transformationsgleichungen reduziert werden

Anstelle der kompletten Matrixmultiplikation benutzen wir für alle Koordinaten X und Y einfach diese beiden Transformationsgleichungen, um die Bildschirmkoordinaten X' und Y' zu erhalten.

g) Umwandeln der bildwichtigen Punkte aus dem Weltsystem in das Bildschirmsystem.

Erst jetzt sind wir soweit, daß wir tatsächlich zeichnen können. Die Matrixrechnung dient uns also im Vorfeld des eigentlichen Programms dazu, die Elemente r11 bis t2 zu ermitteln. Im Programm aber findet keine Matrixoperation mehr statt, dort wird nur noch mit den fertigen Gleichungen gearbeitet.

```
10 REM ********************************   <060>
20 REM *                              *   <069>
30 REM *      T R A N S - 2 D         *   <236>
40 REM * DEMONSTRATION VON 2D-GRAFIK  *   <125>
50 REM *     HEIMO PONNATH HAMBURG    *   <094>
60 REM *                              *   <109>
70 REM ********************************   <120>
80 IF A=0 THEN A=1:LOAD"HIRES4.OBJ",8,8   <152>
90 SYS 49152:HAN:HOF:REM AKTIVIEREN       <070>
100 REM                                   <162>
110 REM ---- DATEN DES HAUSES ---------   <133>
120 DATA 10,5,10,12,15,18,20,12,20,5      <040>
130 DIM H(2,6):REM FELD FUER HAUSPUNKTE   <034>
140 FOR I=1 TO 5                          <240>
150 :READ H(1,I),H(2,I)                   <210>
160 NEXT I                                <244>
170 H(1,6)=H(1,1):H(2,6)=H(2,1)           <239>
180 REM                                   <242>
190 REM ---- PARAMETER ----------------   <243>
200 XM=319:YM=199                         <074>
210 XU=0:XO=30:YU=0:YO=25                 <134>
220 XC=10:YC=5                            <035>
230 PRINT CHR$(147)                       <003>
240 INPUT"ROTATIONSWINKEL =";W            <084>
250 W=W*π/180:REM UMRECHNUNG IN BOGENMASS <094>
260 INPUT"SKALIERUNGSFAKTOREN SX,SY=";SX,S
    Y                                     <145>
270 REM                                   <078>
280 REM ---- ZEICHENWERTE -------------   <054>
290 ZF=5:HF=0:L=1                         <228>
300 REM                                   <108>
310 REM ---- GESAMTMATRIXELEMENTE -----   <162>
320 RA=XM*SX*COS(W)/(XO-XU):REM DAS IST R1
    1                                     <055>
330 RB=-YM*SX*SIN(W)/(YO-YU):REM R12      <188>
340 RC=-XM*SY*SIN(W)/(XO-XU):REM R21      <102>
350 RD=-YM*SY*COS(W)/(YO-YU):REM R22      <220>
360 T1=XM*(XC*(1-SX*COS(W))+SY*YC*SIN(W)-X
    U)/(XO-XU)                            <012>
370 T2=-YM*(YC*(1-SY*COS(W))-SX*XC*SIN(W)-
    YO)/(YO-YU)                           <211>
380 REM                                   <188>
390 REM ---- TRANSFORMATIONSGLEICHUNGEN   <027>
400 DEF FN X(X)=RA*X+RC*Y+T1              <217>
410 DEF FN Y(Y)=RB*X+RD*Y+T2              <200>
420 REM                                   <228>
430 REM ---- ZEICHNEN -----------------   <025>
440 HAN:LOE:FAR,ZF,HF                     <222>
450 FOR I=1 TO 5                          <040>
460 :X=H(1,I):Y=H(2,I):X1=FN X(X):Y1=FN Y(
    Y)                                    <184>
465 :X=H(1,I+1):Y=H(2,I+1):X2=FN X(X):Y2=F
    N Y(Y)                                <191>
470 :LIN,X1,Y1,X2,Y2,L                    <107>
480 NEXT I                                <054>
490 GET A$:IF A$="" THEN 490              <015>
500 HOF                                   <245>
520 REM                                   <074>
530 REM ---- PROGRAMMENDE -------------   <160>
540 INPUT"NOCH EINMAL (J/N)";A$           <178>
550 IF A$="J" THEN 240                    <132>
560 AUS                                   <024>
570 POKE 52,160:POKE 56,160               <126>
580 CLR                                   <186>
590 END                                   <084>
```

**Listing 27. »TRANS-2D«.
Ein Demo-Programm für 2D-Grafik.**

Ein Beispielprogramm zur 2D-Grafik

Damit Sie unser Haus nun tatsächlich auch in Aktion sehen können, finden Sie in »TRANS-2D« ein Programm (Listing 27), aus dem Sie das Prinzip der 2D-Grafikprogrammierung entnehmen können. Wie Sie aber sicher schon bemerkt haben, ist alles Wichtige – nämlich die Bestimmung des Algorithmus zur Berechnung der Elemente r11 bis t2 – bereits vorher geschehen. Für jede neue Grafikoperation, die nicht eine Rotation mit anschließender Skalierung – bezogen auf einen beliebigen Bezugspunkt – ist, muß diese ganze Kette der Algorithmus-Entwicklung neu durchgegangen werden. Erst die Ergebnisse landen wieder als Gleichungen im Programm.

Alle Parameter sind hier im ersten Programmabschnitt schon festgelegt. Lediglich die Rotation (in Grad) und die Skalierungsfaktoren können Sie noch selbst variieren. Benutzen Sie bitte die Grafikbefehls-Erweiterung »HIRES4. OBJ« für dieses Programm. Nebenbei bemerkt: TRANS-2D wartet geradezu darauf, von Ihnen in irgendeiner Weise verändert zu werden! Sie können beispielsweise noch mehr Parameter zur Eingabe freigeben und deren Wirkungen beobachten. Sie könnten Parameter in einer Schleife verändern lassen. Sie könnten andere Grafikobjekte als das nüchterne Haus verwenden. Ihrer Fantasie sind keine Grenzen gesetzt! Beachten Sie, daß Sie bei der Eingabe der Parameter den Wert »1« nicht überschreiten (am besten Sie wählen Werte zwischen 0,2 bis 0,8). Ansonsten gibt der Computer die Fehlermeldung »Illegal Quantity Error« aus. Diese Fehlermeldung ist allerdings nicht lesbar, da sich der C64 im Hires-Modus befindet.

Wenn Sie genau hingesehen haben, werden Sie außer der Skalierung und der Rotation des Hauses auch noch eine Verzerrung beobachtet haben. Der Grund dafür ist im unterschiedlichen Seitenverhältnis zu suchen, die Weltsystem (30 x 25 Meter) und Bildschirm (320 x 200 Punkte) aufweisen. Dazu wirkt sich noch verstärkend aus, daß wir das Weltsystem um den jeweils eingegebenen Winkel W gedreht haben, wodurch sich ebenfalls die Seitenverhältnisse ändern. Man könnte das mittels der Anpassungsmatrix wieder ausgleichen. Aber hier sollte nur ein kurzer Überblick der Techniken gegeben werden, und deshalb wenden wir uns nun der 3D-Grafik zu.

3D-Grafik

Von der 2D- zur 3D-Grafik ist es kein großer Schritt mehr: Im Prinzip brauchen wir nur alle bisherigen Überlegungen etwas zu ergänzen. So ist ein Punkt im Raum durch drei Koordinatenangaben vollständig definiert.

In Bild 68 sehen Sie den Punkt P1(X1,Y1,Z1) in einem rechtshändigen Koordinatensystem. Erinnern Sie sich: Man kann darin beispielsweise die Y-Achse in die Z-Achse überführen durch Drehen in die Richtung, in die die Finger der offenen rechten Hand weisen, wenn der Daumen in die Richtung der X-Achse zeigt.

Sehen wir uns die Transformationen in der 3D-Grafik an, dann stellen wir fest, daß die gleichen Techniken, die wir von der 2D-Grafik her kennen, hier auch benutzt werden. Alle Transformationsmatrizen kann man als einfache Erweiterungen der entsprechenden 2D-Matrizen verstehen. Anstelle von 3,3-Matrizen werden hier aber 4,4-Matrizen gebraucht, und die Punktmatrix ist keine 1,3-Matrix, sondern eine 1,4-Matrix. Die Erweiterung geschieht wieder über einen Dummy-Wert w, der lediglich der besseren Berechenbarkeit dient:

$(X*w \quad Y*w \quad Z*w \quad w)$

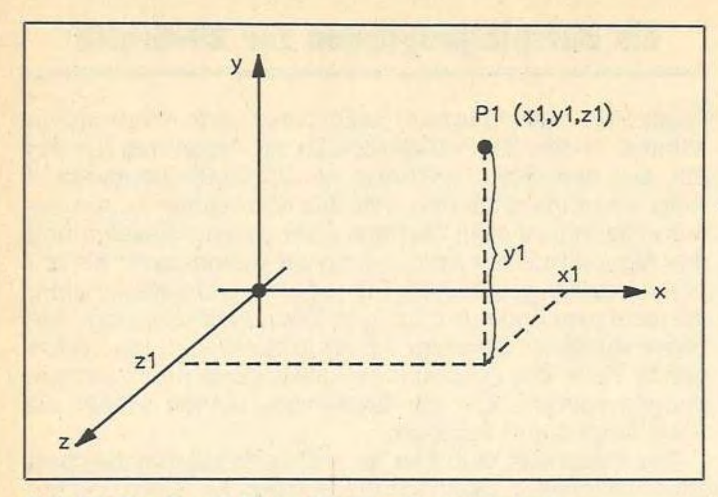


Bild 68. Der Punkt P1 im 3D-Koordinatensystem

3D-Translationen

Wenn T1, T2 und T3 die Verschiebungen in X-, Y- und Z-Richtung bedeuten, dann lautet die Translationsmatrix wie sie in Bild 69 zu sehen ist.

$$T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ T1 & T2 & T3 & 1\end{bmatrix}$$

Bild 69. Die Matrix der 3D-Translation

Sie sehen, es handelt sich einfach um eine Erweiterung der 2D-Translationsmatrix.

3D-Skalierung

Nennen wir Sx, Sy und Sz die Skalierungsfaktoren in der Richtung X, Y und Z, dann gilt für die 3D-Skalierungsmatrix das Bild 70.

$$S=\begin{bmatrix}Sx & 0 & 0 & 0\\ 0 & Sy & 0 & 0\\ 0 & 0 & Sz & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Bild 70. Die 3D-Skalierungsmatrix

Wieder ist die Verwandtschaft zur 2D-Skalierung sofort zu erkennen.

3D-Rotationen

In der 2D-Grafik waren nur Rotationen um eine Drehachse möglich, die durch den Ursprung aus der Bildebene heraus verlief. Wir erkennen diese Achse nun als die Z-Achse. Demzufolge lautet die Rotationsmatrix einer Drehung um die Z-Achse im 3D-Fall entsprechend Bild 71.

$$Rz=\begin{bmatrix}\cos W & \sin W & 0 & 0\\ -\sin W & \cos W & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Bild 71. 3D-Rotation um die Z-Achse

Der Winkel W wird ebenso wie bei der 2D-Grafik als positiv betrachtet, wenn die Drehrichtung – von dem positiven Ende der Z-Achse her gesehen – gegen den Uhrzeigersinn stattfindet.

Wir haben nun aber im Gegensatz zur 2D-Grafik noch zwei weitere Drehachsen zur Verfügung. Bei einer Rotation um die X-Achse sieht die dazugehörende Matrix so aus (Bild 72).

Und bei einer Drehung um die Y-Achse gilt Bild 73.

$$Rx=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos W & \sin W & 0\\ 0 & -\sin W & \cos W & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Bild 72. Die Matrix einer Rotation um die X-Achse

$$Ry=\begin{bmatrix}\cos W & 0 & -\sin W & 0\\ 0 & 1 & 0 & 0\\ \sin W & 0 & \cos W & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Bild 73. 3D-Rotationsmatrix bei Drehung um die Y-Achse

In allen Fällen der Rotation ist W dann positiv, wenn die Drehung – vom positiven Ende der Rotationsachse her gesehen – gegen den Uhrzeigersinn verläuft.

Man kann nun natürlich wieder mehrere dieser einfachen Transformationen miteinander kombinieren. Dann sind die Einzelmatrizen wie im 2D-Fall in der richtigen Reihenfolge miteinander malzunehmen, und es ergibt sich eine kombinierte Matrix. Diese wird dann von folgender Form sein (siehe Bild 74):

$$M=\begin{bmatrix}r_{11} & r_{12} & r_{13} & 0\\ r_{21} & r_{22} & r_{23} & 0\\ r_{31} & r_{32} & r_{33} & 0\\ t_1 & t_2 & t_3 & 1\end{bmatrix}$$

Bild 74. Grundform einer kombinierten 3D-Transformations-Matrix

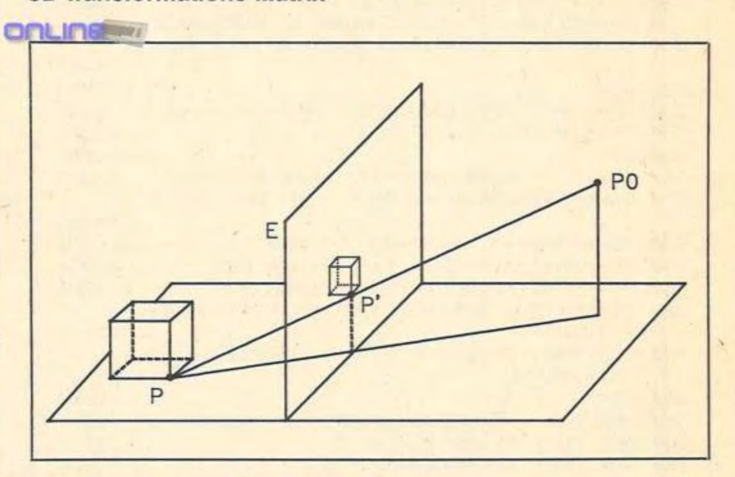


Bild 75. Zum Begriff der Projektion

Im Endeffekt könnte man nun ebenso vorgehen wie in der zweidimensionalen Grafik, nämlich einfach drei Transformationsgleichungen aufstellen, wenn es da nicht noch ein Problem gäbe: Der Bildschirm verfügt nur über zwei Dimensionen, nämlich X und Y.

Projektionen

Nach allen durchgeführten 3D-Transformationen ist das Koordinatensystem, in dem wir uns befinden, immer noch ein dreidimensionales Weltkoordinatensystem. Bevor wir nun eine Anpassungsmatrix anwenden können, um Bildschirmkoordinaten zu erzeugen, müssen wir aus den 3D-Punkten solche erzeugen, die in einer Ebene angeordnet sind, also auch zweidimensional sind wie der Bildschirm. Man nennt solch eine 3D- zu 2D-Transformation eine Projektion, und Bild 75 zeigt Ihnen dafür ein Beispiel.

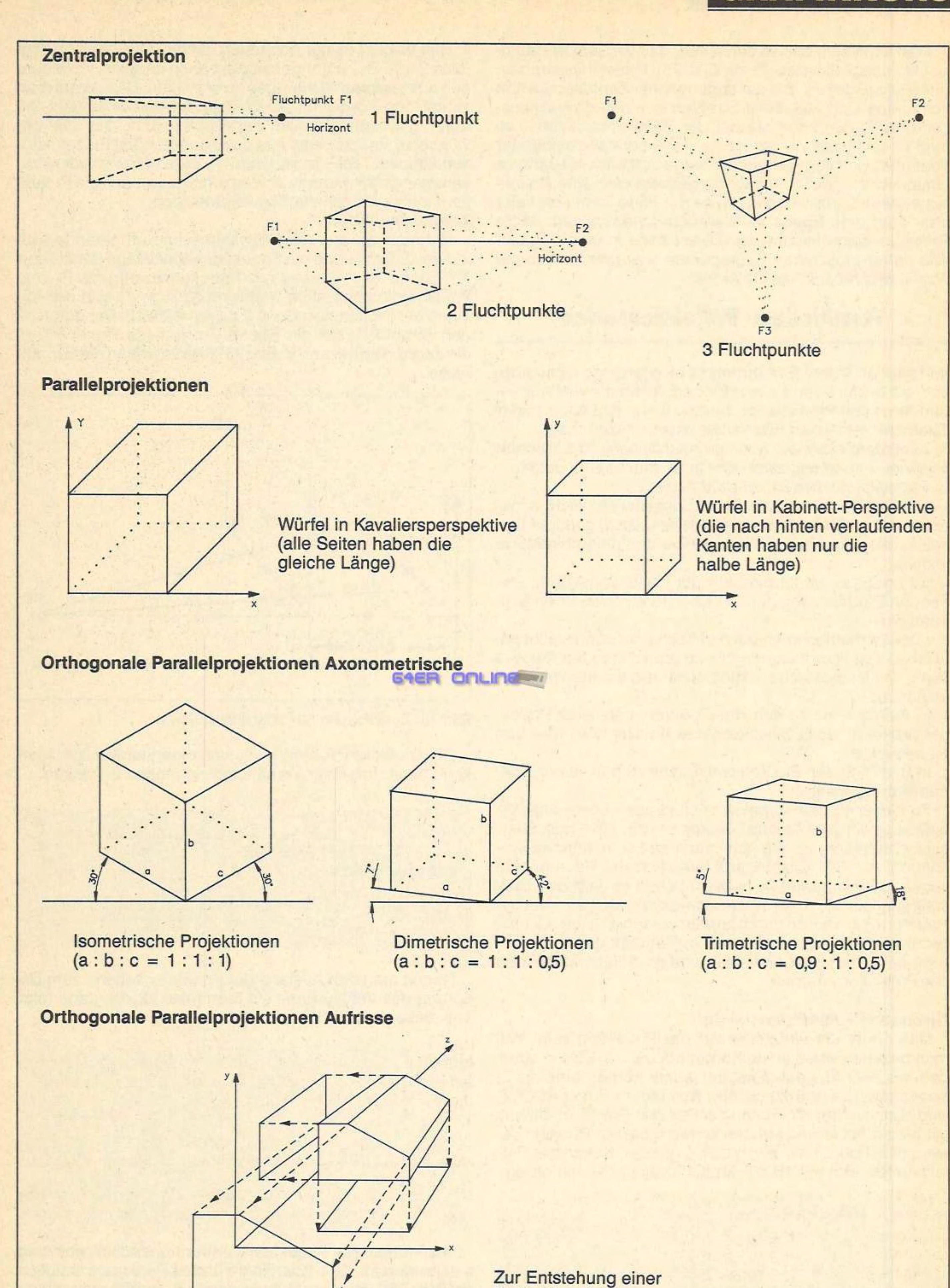


Bild 76. Beispiele für verschiedene Projektionsarten

Man kann sich das so vorstellen, daß von jedem Raumpunkt (beispielsweise P im Bild 75) Projektionsstrahlen (hier P_0) ausgehen. Bei der sogenannten Zentralprojektion schneiden sich alle diese Strahlen in einem Projektionszentrum P_0, während sie bei der Parallelprojektion - so sagt es uns ja schon der Name - alle parallel zueinander verlaufen, dort liegt der gemeinsame Schnittpunkt dann im Unendlichen. Im Strahlengang befindet sich eine Projektionsebene E (die beispielsweise der Bildschirm sein kann oder aber eine Ebene im Weltkoordinatensystem). Jeder Projektionsstrahl schneidet diese Ebene in einem Punkt. Alle derart projizierten Raumpunkte erzeugen das ebene Abbild des räumlichen Objektes.

Verschiedene Projektionsarten

Im Rahmen dieser Einführung ist es allerdings nicht möglich, bis in alle Details einzudringen. Allein die verschiedenen Arten von Projektionen werden Ihnen das Ausmaß der Thematik vor Augen führen. So unterscheidet man:

1. Zentralprojektionen, auch perspektivische Projektionen genannt, mit einem, zwei oder auch drei Fluchtpunkten.
2. Parallelprojektionen auf zwei Arten:

- schräge, bei denen die Projektionsstrahlen nicht senkrecht auf der Projektionsebene stehen. Dazu gehören beispielsweise die Kavaliersperspektive und die Kabinettprojektion.
- orthogonale, mit lotrecht auf der Projektionsebene stehenden Strahlen, von denen es wiederum zwei Arten gibt, nämlich:

a) axonometrische, wobei die Projektionsebene nicht parallel zu den Hauptebenen des zu projezierenden Körpers steht, wie isometrische, dimetrische und trimetrische Projektionen;

b) Aufrisse, mit zu den Hauptebenen parallelen Projektionsebenen, wozu beispielsweise die Dreitafelprojektion zu zählen ist.

In Bild 76 finden Sie Beispielskizzen zu den verschiedenen Projektionsarten.

Zu dieser Vielfalt kommen noch einige andere Aspekte hinzu (die wir zum Teil auch schon bei der 2D-Grafik überschlagen haben, um die man hier aber kaum mehr herumkommt), nämlich Clippingprobleme (welche Raumpunkte liegen beim projizierten Abbild noch innerhalb des Bildschirmfensters?) und Hinterschneidungsfragen (welche Raumpunkte werden durch andere verdeckt?), um nur einige zu nennen. Wir werden uns im Rahmen dieser Einführung daher nur zwei einfachen und grundlegenden Arten der Projektion widmen.

Orthogonale Parallelprojektion

Das dürfte die einfachste Art der Projektion sein. Will man beispielsweise einen Körper auf der XY-Ebene abbilden, wo Z=0 ist, genügt es, bei jedem Körperpunkt die Z-Koordinate gleich 0 zu setzen. Aus einem Punkt P(X,Y,Z) wird dann auf der XY-Ebene der Punkt P'(X,Y,0). Ähnliches gilt für die Abbildung auf den anderen beiden Ebenen: Jeweils eine Koordinate wird gleich 0 gesetzt. Solch eine Projektion läßt sich mittels der Matrix in Bild 77 durchführen.

Die XY-Ebene

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bild 77. Die Matrix der orthogonalen Parallelprojektion auf die XY-Ebene

Hier ist der Fall der Abbildung auf die XY-Ebene gezeigt (also Z=0). Es ist übrigens damit auch möglich, Abbildungen aus anderen Richtungen zu erzeugen. Dazu dreht man zuerst den Körper im 3D-Weltsystem und unterzieht ihn dann der orthogonalen Parallelprojektion auf die gewünschte Hauptebene. Alle dazu gehörenden Einzelmatrizen können - wie vorhin gelernt - durch Multiplikation zusammengefaßt werden, und so erhält man dann die Projektionsmatrix für schiefe Parallelprojektion.

Zentralprojektion

Nehmen wir an, das Projektionszentrum befände sich auf der Z-Achse und die Projektionsebene läge parallel zur XY-Ebene. Der Abstand des Projektionszentrums P_0 vom Weltkoordinatensystem-Ursprung betrüge z1 und der Abstand der Projektionsebene E davon wäre z2. Der durch einen Strahl PP_0 auf die Ebene E projizierte Punkt P' hat die Koordinaten x' und y'. Bild 78 illustriert Ihnen die Verhältnisse.

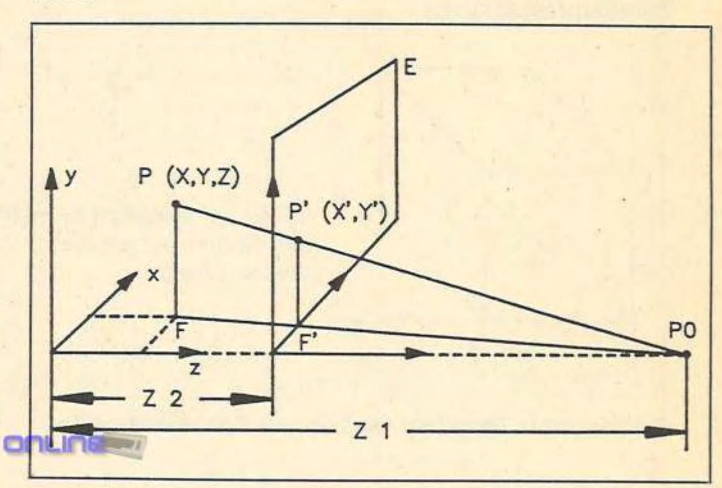


Bild 78. Zum Prinzip der Zentralprojektion

Die Dreiecke PF_0 und $P'F'_0$ sind einander ähnlich. Man kann daher folgende Verhältnisgleichungen aufstellen:

$$\frac{y'}{z1 - z2} = \frac{y}{z1 - z}$$

und analog dazu

$$\frac{x'}{z1 - z2} = \frac{x}{z1 - z}$$

Nimmt man den Abstand der Projektionsebene zum Ursprung des Weltsystems als 0 an (also z2=0), dann folgt aus diesen beiden Gleichungen:

$$y' = \frac{y}{1 - \frac{z}{z1}}$$

$$x' = \frac{x}{1 - \frac{z}{z1}}$$

Rückt man das Projektionszentrum unendlich weit weg - dann wird z1, die z-Koordinate dieses Punktes, unendlich groß, der Bruch $z/z1$ geht dann gegen 0 - erhält man genau die Verhältnisse der Parallelprojektion. Auch diese Berechnungen lassen sich in einer Projektionsmatrix erfassen (Bild 79 auf Seite 155).

Machen Sie mit!



Diesen Beitrag im 64'er-Sonderheft fand ich besonders gut:

Ausgabe: ________ / ________ Seite: ________

Artikel: ________________________________

____

____

Ich wünsche mir für eine der folgenden Ausgaben folgende Themen:

____

____

____

____

Ich möchte an der redaktionellen Gestaltung mitarbeiten.
Meine Vorschläge:

____

____

____

____

Ich kann folgende(s) Programm(e) zur Veröffentlichung anbieten:

____

____

____

____

Dieses Problem habe ich:

____

____

____

____

____

____

____

____

Schreiben Sie uns!

Ich besitze einen: älteren C64 ___ neuen C64II ___

C128 ___ C128D (im Blechgehäuse) ___

mit Laufwerk(en): 1541(alt) ___ 1541c ___ 1541II ___

1570 ___ 1571 ___ 1581 ___

Ich verwende einen Drucker ___

mit 9 Nadeln ___ 24 Nadeln ___

Zusätzlich besitze ich einen

Amiga ___ PC ___ Atari ST ___ andere ___

Diese Note (1 bis 6, 1 am besten) gebe ich dem

64´er-Sonderheft: ___

Das sollte im 64´er-Sonderheft besser werden:

___

___

___

___

Name: ___

Alter: ___ Jahre

Adresse: ___

___

___

Telefon: ___

Bitte schicken Sie die Mitmachkarte
in einem Briefumschlag
an folgende Adresse:

Markt & Technik Verlag AG
Redaktion Sonderhefte
Stichwort: Mitmachkarte 64´er
Hans-Pinsel-Straße 2
8013 Haar b. München

$$P=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{z1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bild 79. Die Matrix einer Zentralprojektion

Die Multiplikation einer Punktematrix mit dieser Projektionsmatrix ergibt allerdings noch nicht die endgültigen ebenen Punktekoordinaten, sondern:

$$(x \quad y \quad 0 \quad (w*z1 - z)/z1)$$

Damit die vierte Koordinate den Dummy-Wert 1 annimmt, müssen alle Koordinaten noch durch diesen vierten Wert dividiert werden (was einer Skalierung entspricht). Auf diese Weise folgt für die Koordinaten eines projizierten Punktes (wobei w=1 gesetzt wurde):

$$\left(\frac{x * z1}{z1 - z} \quad \frac{y * z1}{z1 - z} \quad 0 \quad 1\right)$$

Auch hier kann man perspektivische Abbilder aus anderen Raumrichtungen und in andere Ebenen durch vorheriges Drehen und Verschieben des Körpers erhalten. Alle Einzelmatrizen werden dann wieder zu einer Gesamtmatrix multipliziert.

Erst jetzt kann man auf dieses zweidimensionale Abbild die Matrix zur Bildschirmanpassung anwenden, um schließlich wieder zu Punkten in Bildschirmkoordinaten zu gelangen, die direkt zu zeichnen sind. Wie zuvor im 2D-Fall kann man aber auch hier alle aufeinanderfolgenden Operationen zu einer einzigen Matrix zusammenfassen, aus der dann drei Transformationsgleichungen folgen.

Unsere Wanderung durch die Anfangsgründe der 2D- und der 3D-Grafik ist damit beendet. Falls Sie mehr über dieses faszinierende Gebiet wissen möchten, dann empfehlen wir Ihnen die Bookware »GIGA-CADPlus«, mit der Computergrafiken von besonderer Räumlichkeit geschaffen werden können. (H. Ponnath/Harald Beiler/ag)

»Zylindrisches« Scrollen mit dem Hi-Scroller

Zum Scrollen von Hires-Grafiken gibt es bereits viele Routinen – nicht aber zum zylindrischen Scrollen! Für die Programmierung professioneller Spiele und das Zentrieren von Bildausschnitten ist diese Routine unverzichtbar.

Viele Leser werden sich sicherlich fragen: »Was, um alles in der Welt, ist zylindrisches Scrollen?« Nun, mit »Hi-Scroller« sind Sie jetzt in der Lage, eine hochauflösende Grafik nicht nur »aus dem Bild« hinaus, sondern gleichzeitig wieder hinein zu schieben. Das heißt, wenn Sie mit den entsprechenden Befehlen das Bild beispielsweise nach links verschieben, fließt der herausgeschobene Teil auf der rechten Seite wieder in das Bild hinein. Sie können also mit Hi-Scroller aus simplen Hires-Grafiken ausgezeichnet bewegte Hintergrundgrafiken für Spiele erzeugen. Ebenfalls eignet sich das Programm hervorragend zum Zentrieren von Bildausschnitten. (Für diesen Anwendungszweck wurde es ursprünglich geschrieben.)

Hi-Scroller ist vollständig in Maschinensprache geschrieben und daher sehr schnell. Geben Sie bitte Listing 1 mit dem MSE ein. Das Programm wird mit

```
LOAD "HI*",8,1  <RETURN>
```

von Diskette geladen. (Das Arbeiten mit der Datasette ist ebenfalls problemlos möglich.) Zum Vermeiden eines OUT OF MEMORY ERRORs geben Sie bitte NEW <RETURN> ein. Ab jetzt stehen Ihnen folgende Befehle zur Verfügung:

```
SYS 49152  (Scrollen nach rechts)
SYS 49155  (Scrollen nach links)
SYS 49158  (Scrollen nach oben)
SYS 49161  (Scrollen nach unten)
```

Um einen sichtbaren Effekt zu erzielen, müssen Sie natürlich vorher mit den entsprechenden POKE-Befehlen oder von Ihrem eigenen Programm aus den Grafikbildschirm ab Adresse 8192 ($2000 bis $3F40) eingeblendet haben. Dieser Bereich wird beispielsweise von Hi-Eddi+ und Printfox sowie diversen Grafik- und Zeichenprogrammen benutzt.

Gescrollt wird jeweils um ein Pixel in die gewünschte Richtung. Beim Programmieren in Basic benötigt die Ausführung einer FOR-NEXT-Schleife natürlich viel Zeit, wodurch die hohe Geschwindigkeit von Hi-Scroller nur wenig zum Tragen kommt. Wir empfehlen daher, die Programmaufrufe gegebenenfalls von einer eigenen Maschinenroutine erledigen zu lassen.

Bitte beachten Sie beim Programmieren, daß die Routine ab $C000 im RAM steht und die Zeropage-Adressen 250 bis 254 ($FA bis $FE) als Zähler benutzt. Den Quellcode finden Sie übrigens im Profi-Ass-Format auf der Programmservice-Diskette zu diesem Heft.

Und jetzt viel Vergnügen beim zylindrischen Scrollen hochauflösender Grafiken! (Stefan Bartnitzky/ef)

```
Name : hi-scroller        c000 c21d
-----------------------------------
c000 : 4c 0c c0 4c 84 c0 4c f6  79
c008 : c0 4c 8b c1 a9 00 85 fc  b4
c010 : a2 20 86 fd e8 86 fb a9  2a
c018 : 38 85 fa a9 08 85 14 a9  57
c020 : 19 85 15 a0 00 a2 28 b1  6e
c028 : fa 85 fe 66 fe b1 fc 85  ee
c030 : fe 6a 91 fc a5 fc 18 69  dd
c038 : 08 85 fc a5 fd 69 00 85  2d
c040 : fd ca d0 e7 c6 fd a5 fc  c0
c048 : 38 e9 3f 85 fc a5 fd e9  be
c050 : 00 85 fd e6 fa d0 02 e6  7b
c058 : fb c6 14 d0 c6 e6 fb a5  b4
c060 : fa 18 69 38 85 fa a5 fb  86
c068 : 69 00 85 fb e6 fd a5 fc  a1
c070 : 18 69 38 85 fc a5 fd 69  c3
c078 : 00 85 fd a9 08 85 14 c6  7a
c080 : 15 d0 a0 60 a9 00 85 fa  d8
c088 : a2 20 86 fb e8 86 fd a9  69
c090 : 38 85 fc a9 08 85 14 a9  4f
c098 : 19 85 15 a0 00 a2 28 b1  e6
c0a0 : fa 85 02 26 02 b1 fc 85  4f
c0a8 : 02 2a 91 fc a5 fc 38 e9  ba
c0b0 : 08 85 fc a5 fd e9 00 85  a9
c0b8 : fd ca d0 e7 a5 fc 18 69  c1
c0c0 : 41 85 fc a5 fd 69 01 85  f2
c0c8 : fd e6 fa d0 02 e6 fb c6  e6
c0d0 : 14 d0 c8 a5 fc 18 69 38  da
c0d8 : 85 fc a5 fd 69 01 85 fd  b5
c0e0 : a5 fa 18 69 38 85 fa a5  1c
c0e8 : fb 69 01 85 fb a9 08 85  c1
c0f0 : 14 c6 15 d0 a6 60 a2 00  bf
c0f8 : 86 fa e8 86 fc a9 20 85  af
c100 : fb 85 fd a9 c7 85 14 a9  bf
c108 : 1e 85 15 a9 28 85 fe a2  53
c110 : 18 a0 00 b1 fa 48 b1 fc  61
c118 : 91 fa c8 c0 07 d0 f7 a5  93
c120 : fa 18 69 40 85 fa a5 fb  47
c128 : 69 01 85 fb a5 fc 18 69  68
c130 : 40 85 fc a5 fd 69 01 85  61
c138 : fd a5 14 18 69 40 85 14  e7
c140 : a5 15 69 01 85 15 a0 00  6e
c148 : b1 fa 91 14 b1 fc 91 fa  9c
c150 : c8 c0 07 d0 f7 ca d0 c7  fd
c158 : 88 68 91 fc a5 fa 38 e9  ff
c160 : f8 85 fa a5 fb e9 1d 85  1d
c168 : fb a5 fc 38 e9 f8 85 fc  f2
c170 : a5 fd e9 1d 85 fd a5 14  39
c178 : 38 e9 f8 85 14 a5 15 e9  2a
c180 : 1d 85 15 c6 fe d0 01 60  b9
c188 : 4c 0f c1 a2 00 86 fa e8  12
c190 : 86 fc a9 3e 85 fb 85 fd  11
c198 : a9 3a 85 14 a9 3f 85 15  17
c1a0 : a9 28 85 fe a0 06 a2 18  93
c1a8 : b1 fc 48 b1 fa 91 fc 88  61
c1b0 : 10 f9 a5 fa 38 e9 40 85  64
c1b8 : fa a5 fb e9 01 85 fb a5  38
c1c0 : fc 38 e9 40 85 fc a5 fd  2d
c1c8 : e9 01 85 fd a5 14 38 e9  03
c1d0 : 40 85 14 a5 15 e9 01 85  3c
c1d8 : 15 a0 06 b1 fc 91 14 b1  05
c1e0 : fa 91 fc 88 10 f9 ca d0  91
c1e8 : c9 68 a0 00 91 fa a5 fa  8b
c1f0 : 18 69 08 85 fa a5 fb 69  0f
c1f8 : 1e 85 fb a5 fc 18 69 08  d3
c200 : 85 fc a5 fd 69 1e 85 fd  c6
c208 : a5 14 18 69 08 85 14 a5  33
c210 : 15 69 1e 85 15 c6 fe d0  37
c218 : 01 60 4c a4 c1 ff 00 00  0d
```

Listing 1. »Hi-Scroller« bitte mit dem MSE eingeben

Tips & Tricks

Nützliche Tips und Tricks sind gerade für den Einsteiger, aber auch für Fortgeschrittene, unentbehrlich. Im folgenden finden Sie vier interessante Tips, die Sie zum Ausprobieren anregen sollen.

C64 mit 1,25 MHz Taktfrequenz

Wer hat sich nicht schon über die ewig langen Wartezeiten bei rechenintensiven Fraktalprogrammen oder Programmen wie Giga-CAD geärgert. Besitzer eines C 128 waren da schon immer im Vorteil, da sie ihren Computer auf 2 MHz Taktfrequenz umschalten und so eine erheblich höhere Arbeitsgeschwindigkeit erhalten können. Etwas ähnliches geht auch mit dem C 64.

Wer seinen C 64 öffnet, findet in der Nähe des Video-Chips (der große IC im Blechgehäuse) eine Drahtbrücke, die zur Umschaltung von der PAL- auf die NTSC-Fernsehnorm dient. Was bewirkt diese Drahtbrücke? Steht sie auf PAL (in neueren C 64 ist sie geschlossen; in älteren am Lötpunkt PAL angelötet), so wird die Frequenz des Schwingquarzes durch 2,25 geteilt. Steht sie auf NTSC (in neueren C 64 ist sie nicht geschlossen; in älteren am Lötpunkt NTSC angelötet), so wird die Quarzfrequenz nur durch 1,75 geteilt. Mit dieser Brücke wird der C 64 an die amerikanische Fernsehnorm angepaßt.

Besitzt man nun einen deutschen C 64 (PAL) und stellt die Lötbrücke auf NTSC, so wird die Frequenz logischerweise nicht mehr durch 2,25, sondern durch 1,75 geteilt. Der Prozessor läuft dann anstatt mit 0,98 MHz mit 1,25 MHz. Das bewirkt eine Geschwindigkeitssteigerung um fast 30 Prozent. Für die Bausteine im C 64 ist dies nicht

Druckeranpassungen gesucht

Gute Programme, die mit Druckausgaben arbeiten, bieten zwar oft eine Anpassung an verschiedene Drucker. Oft reichen aber die Anpassungen nicht aus, um mit jedem beliebigen Drucker optimal zusammenzuarbeiten.

Wenn Sie selbst bei einem Programm (Textverarbeitung, Malprogramm etc.) eine entsprechende Anpassung neu entwickelt haben, behalten Sie Ihre Ergebnisse nicht für sich. Schicken Sie uns Ihre Änderungen mit genauer Angabe des Druckers und notwendiger Einstellung aller DIP-Schalter. Helfen Sie damit allen Lesern, die ähnliche Probleme haben und bisher zu keiner Lösung gekommen sind.

Ist es Ihnen gelungen, das DTP-Programm »Giga-Publish« an MPS-801/803 und kompatible Drucker anzupassen?

Sind Sie Besitzer eines 24-Nadel-Druckers und wissen, wie der Drucker optimal eingesetzt wird?

Schreiben Sie an folgende Adresse:
Markt & Technik Verlag AG
Redaktion Sonderhefte
Stichwort: Druckeranpassung
Hans-Pinsel-Straße 2
8013 Haar bei München

Wir werden alle Tips zu Druckern sammeln und in einem der nächsten Sonderhefte zusammenstellen.



schädlich, da diese geänderte Frequenz noch im Toleranzbereich liegt. Lediglich der Monitor bzw. Fernseher, die ja auf genaue Einhaltung der Frequenzen abgestimmt sind, funktionieren nicht mehr. Eine Bildschirmanzeige ist also nicht mehr möglich. Auch der Diskettenbetrieb oder Kassettenbetrieb ist nicht mehr möglich. Jedoch laufen alle Programme, die Berechnungen durchführen, um fast 30 Prozent schneller.

Will man nun also in den Genuß eines schnellen C64 kommen, so braucht man nur die Drahtbrücke durch einen Schalter zu ersetzen. Bei Berechnungen wird dieser Schalter dann auf die schnelle Betriebsart gesetzt. Soll der Computer etwas laden, speichern oder auf dem Bildschirm darstellen, ist mit dem Schalter die »langsame« Betriebsart einzustellen. Vier Stunden Apfelmännchenberechnung lassen sich so auf knapp drei Stunden Wartezeit verkürzen, ohne daß es zu Problemen kommt.

(Andreas Beermann/ag)

PEEKs & POKEs für Grafik

(VIC=Speicherzelle 53248)

Grafikmodus
Einschalten: POKE VIC+17,PEEK(VIC+17) OR 32
Ausschalten: POKE VIC+17,PEEK(VIC+17) AND 223

Grafikspeicher
Grafikspeicher-Startadresse ab Adresse 8192: POKE VIC+24,PEEK(VIC+24) OR 8

Grafik löschen
Grafikspeicher löschen (Voraussetzung: Beginn ab 8192)
FOR I=8192 TO 8192+8000:POKE I,0:NEXT I

Multicolor-Modus
Einschalten (Voraussetzung: Grafikmodus wurde zuvor eingeschaltet) POKE VIC+22,PEEK(VIC+22) OR 16

1. Grafikspeicher schützen. Wie Sie die Hires-Grafik ein-/ausschalten, den Beginn des Grafikspeichers auf Adresse 8192 legen und den Grafikspeicher löschen, finden Sie in der zugehörigen Abteilung unserer Liste.

Anschließend liegt der Grafikspeicher jedoch mitten im eigentlich für das Basic-Programm und die Variablen verwendeten Speicherbereich. Eine Möglichkeit, dieses Problem zu lösen, besteht darin, den für Basic-Programme verfügbaren Speicherbereich zu begrenzen und dem C64 mitzuteilen, daß er bei Adresse 8191 endet:

POKE 5,255:POKE 56,31

Diese beiden Befehle sollten sich in der ersten Zeile Ihres Grafikprogramms befinden! Ihnen stehen nun zwar nur noch 7 KByte für Ihr Programm zur Verfügung, zum Experimentieren mit der Grafik reicht das jedoch allemal.

2. Punkte setzen/löschen: Die »Grafik-POKEs« übernehmen zwar alle notwendigen Vorbereitungen, der eigentliche Sinn von Grafikprogrammen, das Setzen oder Löschen von Punkten, fehlt jedoch in der Liste. Die folgenden beiden Formeln setzen voraus, daß den Variablen X und Y die Punktkoordinaten zugewiesen wurden.

AD = 320 * INT(Y/8) + (Y AND 7) + 8*INT(X/8)
BN = 7 - (X AND 7)

Nach diesen Berechnungen kann der betreffende Punkt gesetzt oder gelöscht werden.

Setzen: POKE 8192+AD,PEEK(8192+AD) OR 2↑BN
Löschen: POKE 8192+AD,PEEK(8192+AD) AND (255-2↑BN)

Die obigen Berechnungen funktionieren jedoch nur, wenn die Hires-Grafik mit 320 x 200 Punkten eingeschaltet ist. AD ist dabei die Nummer des Bytes ab der Anfangsadresse des Grafikbildschirms und BN ist die Nummer des Bits in dem Byte Anfangsadresse + AD.

PEEKs & POKEs für Sprites

(VIC=53248, NR=Spritenummer (0 bis 7))

Breite verdoppeln
POKE VIC+29,PEEK(VIC+29) OR 2↑NR

Höhe verdoppeln
POKE VIC+23,PEEK(VIC+23) OR 2↑NR

Kollision feststellen
PEEK(VIC+30):POKE VIC+30,0
(die Bits der beiden kollidierten Sprites sind gesetzt; in VIC+30 muß (!) anschließend der Wert 0 gePOKEt werden, da diese »Kollisionsspeicherzelle« nicht automatisch gelöscht wird!).

Multicolor-Sprite
Einschalten: POKE VIC+28,PEEK(VIC+28) OR 2↑NR

Spritefarben
Definieren: POKE VIC+39+NR,Farbcode

Sprites einschalten
POKE VIC+21,PEEK(VIC+21) OR 2↑NR

Sprites ausschalten
POKE VIC+21,PEEK(VIC+21) AND 255-2↑NR.

X-/Y-Koordinaten
Festlegen durch POKE VIC+2*NR,X:POKE VIC+2*NR +1,Y

Die »Sprite-POKEs« werden verständlicher, wenn Sie ein wenig über den Umgang des C64 mit Sprites erfahren. Bis zu acht Sprites können gleichzeitig verwaltet werden. Jedes Sprite bekommt eine Nummer zwischen 0 und 7. Für eine Sprite-Funktion ist meist ein Register des VIC-Chips zuständig, beispielsweise Register 29 für die Verdoppelung der Sprite-Höhe. Um ein Register anzusprechen, müssen Sie die Registernummer zur Basisadresse des VIC-Chips addieren:

53248 (Basisadresse) + 29 (Höhen-Register) = 53277

Die Speicherzelle 53277 beeinflußt also die Sprite-Höhe. Um gezielt ein bestimmtes Sprite ansprechen zu können, ist bei fast allen für die Sprites zuständigen Registern je ein Bit einem Sprite zugeordnet.

Bit 0 Sprite Nummer 0
Bit 1 Sprite Nummer 1
...
...
Bit 7 Sprite Nummer 7

Um nun gezielt die Höhe von Sprite Nummer 5 zu verdoppeln, ohne auch alle anderen Sprites zu beeinflussen, muß Bit 5 von Register 53248 gesetzt werden. Gesetzt bzw. gelöscht werden Bits mit Hilfe der logischen Operatoren OR (Setzen) und AND (Löschen). Das folgende Schema zeigt, wie einzelne Bits gezielt manipuliert werden:

Bit NR setzen POKE X,PEEK(X) OR 2↑NR
Bit NR löschen POKE X,PEEK(X) AND 255-(2↑NR)

Zur Verdoppelung der Höhe eines Sprites finden Sie in der Tabelle den POKE:

POKE VIC+29,PEEK(VIC+29) OR 2↑NR

NR ist hierbei die Spritenummer (0 bis 7), VIC die Basisadresse des VIC-Chips (53248). Um die Höhe von Sprite 3 zu verdoppeln, setzen Sie einfach das entsprechende Bit Nummer 3 mit der Anweisung

POKE 53248+29,PEEK(53248+29) OR 2↑3

Die Verdoppelung können Sie jederzeit rückgängig machen, indem Sie mit der folgenden Anweisung dieses Bit wieder löschen:

POKE 53248+29,PEEK(53248+29) AND 255−2↑3

Dieser kleine »Ausflug« sollte verdeutlichen, wie die Inhalte der Sprite-Register vom C64 interpretiert werden. Zum Umgang mit Sprites halten Sie sich bitte einfach an die POKE-Liste. Die Anwendung dürfte problemlos sein, nachdem Ihnen nun die Bedeutung der Variablen VIC und NR bekannt ist. (Said Baloui/ag)

Farbige C128-Grafik an den C64 angepaßt

Der C64 und der C128 im 40-Zeichen-Modus benutzen (üblicherweise) denselben Speicherbereich für Hires-Grafiken: $2000 bis $3FFF. Das gilt aber nur unter dem Gesichtspunkt, ob ein Pixel auf dem Grafikbildschirm ein- oder ausgeschaltet ist. Bei farbigen Hires-Bildern wird es schon komplizierter: Zur Speicherung des Color-RAM verwendet der C128 den Bereich von $1C00 bis $1FE7. Allerdings ist dieser Speicherplatz bei den bekannten Grafikprogrammen des C64 nicht üblich.

Eines der beliebtesten Mal- und Zeichenprogramme für den C64, »Hi-Eddi«, legt die Farbinformation beispielsweise für so ein Color-Hires-Bild unmittelbar hinter die eigentliche Grafik (ab $4000) und speichert den Bereich von $2000 bis $43E7 als Bildfile auf Diskette.

Um nun farbige Hires-Grafiken, die auf dem C128 entwickelt wurden (eine Menge bequem zu handhabender Grafikanweisungen im Basic 7.0 stehen ja dafür zur Verfügung), für ein gutes Mal- und Zeichenprogramm des C64 zugänglich zu machen, dient das Programm »Colorconverter«. Das Programm (Listing 1, bitte mit dem MSE Seite 159 eingeben) ist nur im C64-Modus lauffähig. Geladen wird es mit

```
LOAD"COLORCONVERTER",8
```

und gestartet mit RUN.

Danach befindet sich der Maschinencode im vor Basic geschützten Bereich, von $C000 bis $C2AF.

Das Menü bietet drei Möglichkeiten:

<D> Directory
<G> C128-Farb-Grafikfile von Disk laden
<E> Programmende (es wird ein System-Reset ausgeführt).

Jedes C128-Color-Hiresbild wird nach dem Laden auf dem C64-Hires-Bildschirm angezeigt (das Farb-RAM wurde vorher nach $0400 »geschoben«). Nach Druck auf eine beliebige Taste erscheint das Untermenü:

<S> als »Hi-Eddi«-Farbbild speichern
<M> Rückkehr ins Hauptmenü.

An den Filenamen des C64-Colorbildes, das aus dem C128-Grafikfile erzeugt wurde, hängt das Programm als Endung noch ».64« (zur Vermeidung von gleichen Filenamen und zur besseren Unterscheidung) an. Es kann dann jederzeit zur weiteren Bearbeitung (verändern, invertieren, ausdrucken etc.) in »Hi-Eddi« geladen werden.

Wenn das Programm mit <RUN/STOP RESTORE> oder durch einen »Reset« beendet wurde, läßt es sich jederzeit mit SYS 49152 neu initialisieren. (Harald Beiler/ag)

```
Name : colorconverter    0801 0af1
----------------------------------
0801 : 0b 08 c5 07 9e 32 30 36  0b
0809 : 31 00 00 00 ea ea 78 a9  76
0811 : 40 85 fc a9 08 85 fd a9  80
0819 : 00 85 fe a9 c0 85 ff a2  4e
0821 : 02 a0 00 b1 fc 91 fe c8  93
0829 : d0 f9 e6 fd e6 ff ca d0  aa
0831 : f2 b1 fc 91 fe c8 c0 b0  08
0839 : d0 f7 58 4c 00 c0 ea ae  b4
0841 : 11 d0 ac 18 d0 86 f7 84  13
0849 : f8 a0 00 a2 20 84 f9 86  01
0851 : fa a9 0f 8d 20 d0 8d 21  96
0859 : d0 8c 86 02 20 44 e5 20  4d
0861 : c9 c1 a9 d4 a0 c1 20 1e  e5
0869 : ab 20 c9 c1 a9 0e a0 c2  e2
0871 : 20 1e ab 20 d9 c0 c9 12  7e
0879 : f0 43 c9 1a f0 07 c9 0e  4b
0881 : d0 f1 4c e2 fc a9 4e a0  51
0889 : c2 20 1e ab a9 00 85 c6  97
0891 : a9 01 85 cc a2 00 20 cf  00
0899 : ff 9d 9a c2 e8 c9 0d d0  18
08a1 : f5 86 b7 a2 9a a0 c2 86  e2
08a9 : bb 84 bc 20 ee c0 20 c4  d8
08b1 : c0 20 d9 c0 20 e0 c0 20  5c
08b9 : 84 c0 4c 1d c0 20 44 e5  3e
08c1 : 4c 5d c1 a9 69 a0 c2 20  48
08c9 : 1e ab 20 d9 c0 c9 0d f0  70
08d1 : 07 c9 24 d0 f5 4c 1d c0  98
08d9 : 20 0a c1 a2 00 a0 1c 86  45
08e1 : fb 84 fc 86 fd a0 04 84  2c
08e9 : fe a0 00 b1 fb 91 fd c8  43
08f1 : c0 00 d0 04 e6 fc e6 fe  56
08f9 : a5 fc c9 1f d0 ed c0 e9  c6
0901 : d0 e9 60 20 9c c0 ad 11  8b
0909 : d0 09 20 8d 11 d0 ad 18  96
0911 : d0 29 f0 09 08 8d 18 d0  c2
0919 : a5 cb c9 40 f0 fa 60 a6  d4
0921 : f7 a4 f8 8e 11 d0 8c 18  74
0929 : d0 20 44 e5 60 a9 01 a2  74
0931 : 08 a0 03 20 ba ff a5 b7  00
0939 : a6 bb a4 bc 20 bd ff a9  c1
0941 : 00 a2 00 a0 1c 20 d5 ff  c1
0949 : 60 20 4e c1 20 aa c0 a6  2d
0951 : b7 ca a9 20 9d 9a c2 a2  db
0959 : 0d a0 00 b9 ac c2 9d 9a  7a
0961 : c2 e8 c8 c0 03 d0 f4 a9  bf
0969 : 01 a2 08 a0 03 20 ba ff  ee
0971 : a9 10 a6 bb a4 bc 20 bd  6f
0979 : ff a9 f9 a2 ff a0 43 20  72
0981 : d8 ff a2 0f a9 20 9d 9a  2b
0989 : c2 ca d0 f8 60 a2 00 86  2c
0991 : fb 86 fd a0 1c 84 fc a0  7e
0999 : 40 84 fe 60 a9 01 a2 ab  6b
09a1 : a0 c2 20 bd ff a2 08 a0  d9
09a9 : 60 84 b9 86 ba 20 d5 f3  76
09b1 : a5 ba 20 b4 ff a5 b9 20  a6
09b9 : 96 ff a9 00 85 90 a0 03  1f
09c1 : 84 02 20 a5 ff 85 09 a4  9d
09c9 : 90 d0 34 20 a5 ff a4 90  e0
09d1 : d0 2d a4 02 88 d0 e9 a6  a5
09d9 : 09 20 cd bd a9 20 20 d2  df
09e1 : ff 20 a5 ff a6 90 d0 17  ba
09e9 : aa f0 06 20 d2 ff 18 90  40
09f1 : f0 a9 0d 20 d2 ff 20 e1  6f
09f9 : ff f0 04 a0 02 d0 c1 20  73
0a01 : 42 f6 20 d9 c0 4c 1d c0  66
0a09 : a2 27 a9 c0 20 d2 ff ca  ef
0a11 : d0 f8 60 20 20 20 20 20  3d
0a19 : 20 20 20 20 43 4f 4c 4f  d4
0a21 : 52 43 4f 4e 56 45 52 54  34
0a29 : 45 52 20 31 32 38 2f 36  d4
0a31 : 34 0d 20 20 20 20 20 20  bc
0a39 : 20 20 20 28 43 29 20 36  e1
0a41 : 34 27 45 52 20 2f 20 48  31
0a49 : 2e 42 2e 0d 00 0d 20 20  ef
0a51 : 12 20 44 20 92 49 52 45  d0
0a59 : 43 54 4f 52 59 0d 0d 20  57
0a61 : 20 12 20 47 20 92 52 41  de
0a69 : 46 49 4b 20 4c 41 44 45  95
0a71 : 4e 20 28 43 31 32 38 29  1a
0a79 : 0d 0d 20 20 12 20 45 20  90
0a81 : 92 4e 44 45 20 28 52 45  0b
0a89 : 53 45 54 29 00 93 0d 46  16
0a91 : 49 4c 45 4e 41 4d 45 20  ef
0a99 : 44 45 52 20 31 32 38 2d  f8
0aa1 : 47 52 41 46 49 4b 3a 00  02
0aa9 : 93 12 20 53 20 92 50 45  1a
0ab1 : 49 43 48 45 52 4e 20 43  f5
0ab9 : 36 34 20 28 48 49 2d 45  25
0ac1 : 44 44 49 2d 46 4f 52 4d  e2
0ac9 : 41 54 29 0d 0d 12 20 4d  9d
0ad1 : 20 92 45 4e 55 45 0d 0d  23
0ad9 : 00 58 20 20 20 20 20 20  d5
0ae1 : 20 20 20 20 20 20 20 20  e1
0ae9 : 20 00 24 2e 36 34 00 00  dd
```

Listing 1. Das Maschinenprogramm »Colorconverter« wird wie ein Basic-Programm geladen und gestartet

Checksummer V3 und MSE

Diese beiden Programme sind unentbehrlich beim Abtippen unserer Listings. Sie helfen, Tippfehler vor allem bei Maschinenprogrammen zu vermeiden und sparen eine Menge Zeit.

Nobody is perfect. Jeder Computer-Fan, egal ob blutiger Anfänger oder ausgefuchster Profi, macht beim Abtippen von Programmen Tippfehler. Diese Fehler später zu finden, kann ein langwieriges Unterfangen sein.

Deshalb haben wir für Sie die Programme »Checksummer V3« und »MSE« (MaschinenSpracheEditor) entwickelt. Der Checksummer ist für Basic-Programme und der MSE für Maschinensprache-Listings zuständig.

Der Checksummer

Zuerst einmal müssen Sie das Checksummer-Programm (siehe Listing 1) abtippen. Dabei sollten Sie äußerst sorgfältig vorgehen, vor allem bei den Zahlen in den DATA-Zeilen 20 bis 30. Wenn Sie trotzdem noch einen Tippfehler gemacht haben, meldet sich das Programm später mit einem entsprechenden Hinweis. Wenn Sie fertig sind, speichern Sie das Programm auf Diskette oder Kassette.

Jetzt geht es los:

1. Starten Sie den Checksummer durch die Eingabe von »RUN« und das Drücken der RETURN-Taste.
2. Wenn die Meldung »Checksummer aktiviert...« auf dem Bildschirm erscheint, haben Sie keinen Tippfehler gemacht und der Checksummer ist nun eingeschaltet.
3. Zum Löschen des Basic-Programms geben Sie bitte »NEW« ein. Keine Angst, der Checksummer selbst wird dadurch nicht gelöscht.
4. Nun können wir den Checksummer testen. Geben Sie bitte folgende Zeile ein und drücken Sie die RETURN-Taste: 1 REM

In der linken oberen Bildschirmecke sehen Sie nun die Prüfsumme über die eben eingegebene Basic-Zeile. Sie muß <63> lauten. Dem Checksummer ist es übrigens egal, ob Sie »1 REM« oder »1REM« eintippen. Nur innerhalb von Anführungszeichen ist die richtige Anzahl an Leerzeichen wichtig. Diese Prüfsummen erscheinen (sofern Sie den Checksummer eingeschaltet haben) immer dann, wenn Sie eine Basic-Zeile eintippen und dann die RETURN-Taste drücken. In der 64'er finden Sie die Prüfsumme immer am Ende jeder Programmzeile.

```
10 PRINT"CHECKSUMMER FUER C 64"
11 PRINT:PRINT"EINEN MOMENT, BITTE ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A:NEXT I
13 IF PS<>5765 THEN PRINT"TIPPFEHLER IN DE
   N ZEILEN 20 BIS 22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A:NEXT I
15 IF PS<>16147 THEN PRINT"TIPPFEHLER IN D
   EN ZEILEN 22 BIS 30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"CHECKSUMMER AKTIVIERT."
18 PRINT:PRINT" AUSSCHALTEN : POKE1,55 ODE
   R"SPC(27)"<RUN/STOP+RESTORE>"
19 PRINT:PRINT" ANSCHALTEN : POKE1,53"
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
© 64'er
```

Listing 1. Der »Checksummer 64 V3« für Basic-Listings


```
5 PRINT CHR$(14)                             <242>
10 PRINT"{CLR}"                              <254>
20 PRINT"Q******************************W"   <130>
30 PRINT"{4DOWN,2SPACE}TEST{SPACE,BLUE,6SP
   ACE}"                                     <022>
40 PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"   <108>
© 64'er
```

Bild 1. Die Bedeutung der Steuerzeichen wird im nachfolgenden Text erklärt

In Zeile 10 müssen Sie nach den Anführungszeichen die Tasten <SHIFT CLR/HOME> drücken und nicht die Klammern mit dem Wort CLR eingeben. In Zeile 20 drücken Sie nach den Anführungszeichen die CBM-Taste und den Buchstaben <Q>, gefolgt von mehreren SHIFT- und Stern-Tasten und zum Schluß die CBM-Taste und den Buchstaben <W>. In Zeile 30 ist es viermal die CURSOR-abwärts-Taste, gefolgt von zweimaliger Leertaste, dann <SHIFT T> und normal EST, zum Schluß noch einmal die Leertaste, die Farbtaste Blau <CTRL 7> und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der CBM-Taste und <B> erzeugt werden.

CTRL steht für Control-Taste, so bedeutet {CTRL+A}, daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift, Cursor unten	{SPACE}	Leertaste	{RVSON}	Control-Taste & 9
{UP}	Shift-Taste & Taste neben rechtem Shift; Cursor hoch	{SHIFT-Space}	Shift-Taste & Leertaste	{RVOFF}	Control-Taste & 0
{CLR}	Shift-Taste & 2. Taste ganz rechts oben	{F1} bis {F8}	Funktionstasten	{ORANGE}	Commodore-Taste & 1
{INST}	Shift-Taste & Taste ganz rechts oben	{RETURN}	Return-Taste	{BROWN}	Commodore-Taste & 2
{HOME}	2. Taste von ganz rechts oben	{BLACK}	Control-Taste & 1	{LIG.RED}	Commodore-Taste & 3
{DEL}	Taste ganz rechts oben	{WHITE}	Control-Taste & 2	{GREY 1}	Commodore-Taste & 4
{RIGHT}	Taste ganz rechts unten	{RED}	Control-Taste & 3	{GREY 2}	Commodore-Taste & 5
{LEFT}	Shift-Taste & Taste unten rechts	{CYAN}	Control-Taste & 4	{LIG.GREEN}	Commodore-Taste & 6
		{PURPLE}	Control-Taste & 5	{LIG.BLUE}	Commodore-Taste & 7
		{GREEN}	Control-Taste & 6	{GREY 3}	Commodore-Taste & 8
		{BLUE}	Control-Taste & 7		
		{YELLOW}	Control-Taste & 8		

Tabelle 1. Die Steuerbefehle in den Listings

Diese Zahlen dürfen Sie NICHT mit abtippen.
Als Beispiel sehen Sie Bild 1. Am rechten Rand jeder Spalte sehen Sie die Prüfsummen in eckigen Klammern.
Damit sind wir beim zweiten wichtigen Punkt: Sehen Sie sich die Zeile 240 von Listing 2 genauer an. Nach dem ersten Anführungszeichen nach dem PRINT-Befehl sehen Sie eine geschweifte Klammer { }. Immer, wenn Sie in einem unserer Listings diese Klammern sehen, dürfen Sie das, was innerhalb der Klammern steht, nicht eintippen. Sie müssen die entsprechende Taste drücken. Beispiel:
10 PRINT "{CLR}"
bedeutet: Nach dem Anführungszeichen die »Bildschirmlöschen«-Taste drücken (<SHIFT CLR/HOME>). In Tabelle 1 sehen Sie eine Zusammenfassung aller möglichen Steuertasten mit dem entsprechenden Klartext.
Weiterhin sehen Sie in Bild 1 (Bedeutung der Steuerzeichen) in Zeile 30 ein unterstrichenes »T« nach der Klammer. Das bedeutet, daß Sie ein »T« zusammen mit der SHIFT-Taste drücken müssen, also <SHIFT T>. Wenn ein Zeichen »überstrichen« ist, müssen Sie dieses zusammen mit der CBM-Taste eingeben. Die CBM-Taste befindet sich ganz links unten auf der Tastatur und hat die Aufschrift »C=«.

```
100 REM DIESES PROGRAMM ERZEUGT DEN           <210>
110 REM MSE V1.1 AUF DISKETTE.                <039>
120 REM BESITZER EINER DATASETTE              <178>
130 REM MUESSEN DIE '8' AM ENDE VON           <145>
140 REM ZEILE 343 IN EINE '1' AENDERN!        <176>
150 REM                                       <212>
230 IF PEEK(44)<>32 THEN PRINT"{CLR}SIE HA
    BEN VERGESSEN, DIE POKES EINZUGE- BEN!
    ":END                                     <050>
240 PRINT"{CLR}";:DIM H(75):FOR I=Ø TO 9      <042>
250 H(48+I)=I:H(65+I)=I+1Ø:NEXT:Z=1ØØØ        <136>
260 FOR I=2Ø48 TO 3755 STEP 2Ø:PRINT"{HOME
    }ICH LESE ZEILE:"Z                        <253>
261 FOR N=Ø TO 19:READ A$:IF LEN(A$)<>2 TH
    EN 9ØØ                                    <062>
262 IF PEEK(63)+PEEK(64)*256<>Z THEN 8ØØ      <011>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))    <199>
280 D=H(H)*16+H(L):S=S+D:POKE I+N,D           <165>
290 NEXT:READ V:IF S<>V THEN 9ØØ              <139>
300 S=Ø:Z=Z+1:NEXT:R=PEEK(2111):H=PEEK(21Ø
    6)                                        <126>
301 POKE 5328Ø,R:POKE 53281,H:POKE 646,R:P
    RINT"{CLR}DIE DATA-ZEILEN SIND FEHLERF
    REI!"                                     <080>
302 PRINT"SIE KOENNEN NUN DIE FARBEN DES M
    SE"                                       <209>
303 PRINT"EINSTELLEN.":PRINT"{2DOWN,SPACE,
    RVSON}DRUECKEN SIE <1>, <2> ODER <9>      <205>
304 PRINT"{DOWN,2SPACE}<1> - RAHMEN-/SCHRI
    FTFARBE                                   <013>
305 PRINT"{2SPACE}<2> - HINTERGRUNDFARBE      <233>
306 PRINT"{DOWN,2SPACE}<9> - FARBEN UEBERN
    EHMEN                                     <158>
307 PRINT"{2DOWN}FARBE <1>:"R:PRINT"FARBE
    <2>:"H                                    <066>
308 GET A:IF A=Ø THEN 3Ø8                     <210>
309 IF A=1 THEN R=(R+1)AND 15                 <098>
310 IF A=2 THEN H=(H+1)AND 15                 <086>
311 IF A=9 THEN 34Ø                           <217>
312 GOTO 3Ø1                                  <034>
340 POKE 21Ø6,H:POKE 2111,R                   <153>
342 POKE 631,19:POKE 632,13:POKE 198,2        <135>
343 PRINT"{CLR}SAVE"CHR$(34)"MSE V1.1"CHR$
    (34)",8                                   <091>
344 POKE 43,1:POKE 44,8:POKE 45,172:POKE 4
    6,14:END                                  <140>
800 PRINT"{CLR,RVSON}SIE HABEN ZEILE"Z"{LE
    FT,SPACE}VERGESSEN:":A=PEEK(646)AND 15    <124>
810 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z-2"-"Z+2:POKE 646,A                      <224>
820 GOTO 92Ø                                  <082>
900 PRINT"{CLR,RVSON}SIE HABEN EINEN TIPPF
    EHLER GEMACHT:":A=PEEK(646)AND 15         <154>
910 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z:POKE 646,A                              <173>
920 POKE 631,19:POKE 632,17:POKE 633,13:PO
    KE 198,3:END                              <126>
1000 DATA ØØ,ØB,Ø8,ØA,ØØ,9E,32,3Ø,36,31,ØØ
     ,ØØ,ØØ,A2,Ø8,A9,36,85,A4,A9, 1247        <119>
1001 DATA Ø8,85,A5,A9,ØØ,85,A6,A9,BØ,85,A7
     ,AØ,ØØ,B1,A4,91,A6,C8,DØ,F9, 2888        <054>
1002 DATA E6,A5,E6,A7,CA,DØ,F2,A9,36,85,Ø1
     ,4C,ØØ,BØ,2Ø,D1,B1,A9,ØØ,8D, 2781        <096>
1003 DATA 21,DØ,A9,ØF,8D,2Ø,DØ,8D,86,Ø2,AØ
     ,B3,A9,74,2Ø,FF,B1,AØ,B3,A9, 2679        <089>
1004 DATA B9,2Ø,FF,B1,AØ,ØØ,2Ø,CF,FF,99,Ø1
     ,Ø2,C8,C9,ØD,DØ,F5,88,FØ,D2, 2912        <217>
1005 DATA CØ,11,9Ø,Ø2,AØ,1Ø,8C,ØØ,Ø2,2Ø,EA
     ,B1,AØ,B3,A9,CF,2Ø,FF,B1,2Ø, 2327        <045>
1006 DATA 8E,B4,85,FC,85,62,2Ø,8E,B4,85,FB
     ,85,61,2Ø,A7,B4,DØ,2Ø,AØ,B3, 2864        <199>
1007 DATA A9,E5,2Ø,FF,B1,2Ø,8E,B4,85,6Ø,2Ø
     ,8E,B4,85,5F,2Ø,A7,B4,DØ,ØA, 2624        <091>
1008 DATA A5,61,C5,5F,A5,62,E5,6Ø,9Ø,Ø6,2Ø
     ,43,B3,4C,3A,BØ,A9,AA,AØ,ØØ, 2379        <167>
1009 DATA EA,EA,E6,FB,DØ,Ø2,E6,FC,2Ø,3F,B2
     ,9Ø,EF,4C,FB,B4,A2,Ø2,86,58, 319Ø        <041>
1010 DATA A9,A6,AØ,9D,2Ø,F2,B1,2Ø,E4,FF,FØ
     ,FB,C9,3Ø,9Ø,ØC,C9,47,BØ,Ø8, 297Ø        <231>
1011 DATA C9,3A,9Ø,ØB,C9,41,BØ,Ø7,C9,14,DØ
     ,ØF,4C,ØB,B1,2Ø,D2,FF,A6,58, 2322        <121>
1012 DATA 95,F7,C6,58,DØ,D2,6Ø,AE,8D,Ø2,FØ
     ,26,C9,ØC,DØ,Ø3,4C,ØB,B6,C9, 2685        <057>
1013 DATA 13,DØ,Ø3,4C,8B,B5,C9,ØD,DØ,Ø3,4C
     ,BA,B4,C9,1Ø,DØ,Ø3,4C,68,B5, 2282        <225>
1014 DATA C9,ØE,DØ,Ø6,2Ø,5F,B4,4C,64,B1,4C
     ,92,BØ,A5,F9,2Ø,Ø2,B1,ØA,ØA, 2132        <208>
1015 DATA ØA,ØA,85,F9,A5,F8,2Ø,Ø2,B1,Ø5,F9
     ,6Ø,C9,3A,9Ø,Ø2,69,Ø8,29,ØF, 195Ø        <092>
1016 DATA 6Ø,A6,59,EØ,Ø8,9Ø,1F,A6,58,EØ,Ø2
     ,BØ,Ø6,2Ø,D2,FF,4C,8E,BØ,C6, 25Ø9        <188>
1017 DATA 59,AØ,14,A9,92,2Ø,F2,B1,CA,DØ,FA
     ,84,57,68,68,4C,8B,B1,A6,D3, 2891        <197>
1018 DATA EØ,Ø8,BØ,Ø3,4C,92,BØ,2Ø,D2,FF,A6
     ,58,EØ,Ø2,9Ø,Ø9,C6,59,2Ø,D2, 2468        <049>
1019 DATA FF,C6,58,DØ,F9,4C,8E,BØ,48,4A,4A
     ,4A,4A,2Ø,59,B1,68,29,ØF,C9, 2419        <035>
1020 DATA ØA,9Ø,Ø2,69,Ø6,69,3Ø,4C,D2,FF,A2
     ,FC,9A,2Ø,D1,B1,2Ø,48,B2,2Ø, 2261        <073>
1021 DATA EA,B1,2Ø,9F,B2,A5,FC,2Ø,4E,B1,A5
     ,FB,2Ø,4E,B1,2Ø,ED,B1,A9,3A, 286Ø        <148>
1022 DATA AØ,2Ø,2Ø,F2,B1,A9,ØØ,85,59,2Ø,8E
     ,BØ,2Ø,ED,B1,A4,59,2Ø,EF,BØ, 253Ø        <233>
1023 DATA 91,FB,C8,84,59,CØ,Ø8,9Ø,EC,2Ø,1Ø
     ,B2,A9,12,2Ø,D2,FF,2Ø,8E,BØ, 2657        <105>
1024 DATA 2Ø,EF,BØ,C5,FF,FØ,ØD,2Ø,43,B3,A9
     ,14,AØ,14,2Ø,F2,B1,4C,A2,B1, 2665        <034>
1025 DATA A9,92,2Ø,D2,FF,2Ø,33,B2,2Ø,EØ,B2
     ,2Ø,3F,B2,9Ø,9F,4C,8B,B5,A9, 2648        <123>
1026 DATA 93,2Ø,D2,FF,A2,ØØ,A9,Ø3,9D,ØØ,D8
     ,9D,ØØ,D9,9D,ØØ,DA,9D,ØØ,DB, 2476        <237>
1027 DATA E8,DØ,EF,6Ø,A9,ØD,2C,A9,2Ø,4C,D2
     ,FF,2Ø,D2,FF,98,4C,D2,FF,2Ø, 2965        <160>
1028 DATA E4,FF,FØ,FB,6Ø,84,5D,85,5C,AØ,ØØ
     ,B1,5C,FØ,Ø6,2Ø,D2,FF,C8,DØ, 31ØØ        <077>
1029 DATA F6,6Ø,A5,FB,85,5A,AØ,ØØ,84,5B,B1
     ,FB,18,65,5A,85,5A,9Ø,Ø2,E6, 26Ø6        <156>
1030 DATA 5B,Ø6,5A,26,5B,C8,CØ,Ø8,9Ø,EC,A5
     ,5A,65,5B,85,FF,6Ø,18,A5,FB, 2467        <219>
1031 DATA 69,Ø8,85,FB,9Ø,Ø2,E6,FC,6Ø,A5,FB
     ,C5,5F,A5,FC,E5,6Ø,6Ø,AØ,B3, 31Ø6        <183>
1032 DATA A9,FB,2Ø,FF,B1,AØ,Ø1,B9,ØØ,Ø2,2Ø
     ,D2,FF,CC,ØØ,Ø2,C8,9Ø,F4,A9, 2692        <098>
1033 DATA 14,ED,ØØ,Ø2,AA,2Ø,ED,B1,CA,DØ,FA
     ,A5,62,2Ø,4E,B1,A5,61,2Ø,4E, 2457        <060>
1034 DATA B1,2Ø,ED,B1,A5,6Ø,2Ø,4E,B1,A5,5F
     ,2Ø,4E,B1,EA,EA,EA,EA,EA,EA, 3122        <190>
1035 DATA EA,EA,24,5E,1Ø,Ø1,6Ø,A9,12,2Ø,D2
     ,FF,A2,28,2Ø,ED,B1,CA,DØ,FA, 27Ø3        <087>
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,BØ,Ø1
     ,6Ø,A9,AØ,85,A4,A9,78,85,A6, 2945        <204>
1037 DATA A9,Ø4,85,A5,85,A7,A2,13,AØ,27,B1
     ,A4,91,A6,88,1Ø,F9,CA,FØ,19, 2671        <208>
1038 DATA 18,A5,A4,69,28,85,A4,9Ø,Ø2,E6,A5
     ,18,A5,A6,69,28,85,A6,9Ø,EØ, 25Ø3        <251>
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
     ,ØF,8D,18,D4,A9,ØØ,8D,Ø5,D4, 2776        <000>
1040 DATA A9,F7,8D,Ø6,D4,A9,11,8D,Ø4,D4,A9
     ,32,8D,Ø1,D4,A9,ØØ,8D,ØØ,D4, 2413        <126>
1041 DATA AØ,8Ø,2Ø,Ø9,B3,A9,1Ø,8D,Ø4,D4,6Ø
     ,A2,FF,CA,DØ,FD,88,DØ,F8,6Ø, 2914        <240>
1042 DATA A9,ØF,8D,18,D4,A9,2D,8D,Ø5,D4,A9
     ,A5,8D,Ø6,D4,A9,21,8D,Ø4,D4, 2385        <119>
1043 DATA A9,Ø7,8D,Ø1,D4,A9,Ø5,8D,ØØ,D4,AØ
```

```
     ,FF,20,09,B3,A9,20,8D,04,D4, 2250      <078>
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20
     ,F0,FF,8A,48,98,48,18,A0,06, 2179      <175>
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF
     ,B1,20,12,B3,20,E4,FF,F0,FB, 2931      <093>
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68
     ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704      <088>
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53
     ,43,48,49,4E,45,4E,53,50,52, 1144      <216>
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54
     ,4F,52,20,0D,0D,20,20,20,20, 1023      <038>
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D
     ,41,4E,4E,20,26,20,44,2E,57, 1128      <206>
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20
     ,20,20,50,52,4F,47,52,41,4D, 1102      <117>
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D
     ,20,20,20,53,54,41,52,54,41, 1073      <095>
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00
     ,0D,0D,20,20,20,45,4E,44,41, 1014      <129>
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20
     ,24,00,92,01,01,50,52,4F,47, 1138      <228>
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20
     ,2A,2A,2A,20,46,41,4C,53,43, 1024      <027>
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20
     ,2A,2A,2A,20,20,92,00,0D,0D, 1058      <098>
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A
     ,2A,00,13,01,20,20,12,44,92, 916       <153>
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54
     ,92,41,50,45,0D,00,13,20,20, 1151      <035>
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45
     ,52,00,20,D1,B1,20,48,B2,A0, 1606      <012>
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC
     ,20,8E,B4,85,FB,C5,61,A5,FC, 3207      <251>
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5
     ,60,B0,19,20,A7,B4,D0,14,60, 2860      <112>
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0
     ,05,85,F8,4C,EF,B0,68,68,20, 2749      <088>
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0
     ,09,20,D1,B1,20,48,B2,4C,0B, 2372      <046>
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4
     ,20,EA,B1,20,0D,B5,24,5E,30, 2042      <120>
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26
     ,20,9F,B2,24,5E,10,09,20,4E, 2435      <198>
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20
     ,3F,B2,90,D7,A0,B4,A9,28,20, 2190      <207>
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00
     ,85,5E,A5,61,85,FB,A5,62,85, 3056      <240>
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E
     ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003      <221>
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1
     ,B1,FB,20,4E,B1,C8,C0,08,90, 2566      <070>
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C
     ,A9,20,20,D2,FF,20,10,B2,A5, 2190      <059>
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA
     ,B1,A9,FF,85,B8,85,B9,A9,04, 3073      <029>
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20
     ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315      <189>
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2
     ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596      <111>
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C
     ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812      <015>
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29
     ,BF,D0,03,4C,FB,B4,A9,01,20, 2577      <201>
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF
     ,B1,20,F9,B1,4C,FB,B4,20,68, 2921      <237>
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1
     ,A2,08,C9,44,F0,06,A2,01,C9, 2717      <213>
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00
     ,E0,01,F0,1A,A9,40,8D,20,02, 2403      <101>
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02
     ,C8,CC,00,02,90,F4,C8,C8,D0, 2182      <127>
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02
     ,D0,F4,98,A2,20,A0,02,4C,BD, 2018      <025>
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6
     ,B9,86,57,A9,01,20,C3,FF,A9, 2800      <022>
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20
     ,B4,FF,A5,B9,20,96,FF,20,A5, 2911      <053>
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5
     ,FF,85,62,20,AB,FF,A5,57,85, 2663      <214>
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5
     ,86,5F,84,60,A5,BA,C9,01,D0, 2639      <131>
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62
     ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300      <120>
1085 DATA 1C,20,ED,B1,CA,D0,FA,60,00,00,00
     ,00,00,00,00,00,00,00,00,00, 1230      <143>
```


Listing 2. Der MSE-Lader

Der MSE

Der MSE dient zur Eingabe von Maschinensprache-Programmen. Als erstes müssen Sie den sogenannten »MSE-Lader« (Listing 2) abtippen. Dieser erzeugt erst das eigentliche MSE-Programm auf Diskette oder Kassette.

Wichtig: Vor dem Eintippen des MSE-Laders müssen Sie unbedingt ein paar Befehle eingeben (ohne Basic-Zeilennummer): POKE 44,32 : POKE 8192,0 : NEW

Jetzt können Sie beginnen, das Listing 2 abzutippen. Der MSE-Lader erkennt zwar, wenn Sie beim Eintippen der DATA-Zeilen einen Fehler gemacht haben, aber wenn Sie ganz sicher gehen möchten, sollten Sie den Checksummer vor dem Eintippen aktivieren. Die Prüfsummen für den MSE-Lader finden Sie am Ende der jeweiligen Programmzeilen.

Wenn Sie das Listing 2 nicht auf einmal abtippen möchten, müssen Sie vor jedem neuen Laden des Programms unbedingt die oben genannte POKE-Zeile eingeben!

Wenn Sie alles richtig gemacht haben und das Programm fehlerfrei abgetippt wurde, speichert es sich nach dem Starten selbst auf Diskette oder Kassette unter dem Namen »MSE V1.0«. Dieses fertige MSE-Programm laden Sie dann bei Bedarf wie ein normales Basic-Programm und starten es mit »RUN«.

So arbeitet man mit dem MSE

Als erstes möchte der MSE den Namen des zu bearbeitenden Programms wissen. Dieser steht in der ersten Zeile unserer MSE-Listings. Dann müssen Sie die Start- und Endadresse des Programms eingeben. Dies sind die letzten beiden vierstelligen Hexadezimalzahlen in der ersten Zeile unserer Listings.

Wenn Sie ein Programm von Diskette oder Kassette laden wollen, um an einer bestimmten Stelle weiterzutippen oder noch eine Korrektur vorzunehmen, geben Sie auf die Frage nach der Startadresse ein »L« ein. Danach müssen Sie <D> oder <T> drücken, je nachdem, ob Sie von Diskette oder Kassette (»tape«) laden möchten. Wenn das Programm unter diesem Namen nicht auf der Diskette vorhanden ist oder ein sonstiger Ladefehler vorlag, meldet sich der MSE mit »I/O-ERROR«. In diesem Fall drücken Sie <RUN/STOP RESTORE> und geben einfach noch einmal »RUN« ein.

Beim Abtippen geben Sie nach und nach die abgedruckten Buchstaben und Zahlen des jeweiligen Listings ohne die Freiräume dazwischen ein. Wenn Sie in einer Zeile einen Tippfehler gemacht haben, meldet sich der MSE sofort mit einem Brummton und der Meldung »EINGABE-FEHLER«. Nach einem Druck auf die RETURN-Taste können Sie mit der DEL-Taste den Fehler korrigieren. Wenn Sie das gewünschte Programm vollständig eingegeben haben, speichert es der MSE automatisch auf Diskette oder Kassette.

Bei längeren Listings ist es unwahrscheinlich, daß Sie das komplette Programm auf einmal eingeben. Sie können Ihre bisherige Tipparbeit jederzeit durch <CTRL S> auf Diskette oder Kassette speichern und Ihr Werk später fortsetzen. Sie sollten sich dann allerdings im Heft markieren, wie weit Sie beim Abtippen gekommen sind! Später geben Sie dann nach dem Laden des ersten Programmteils <CTRL N> ein und auf die dann folgende Frage nach der Startadresse die Zeilennummer (Adresse), bei der Sie aufgehört haben zu tippen.

<CTRL M> erlaubt Ihnen jederzeit, Ihr Werk listen zu lassen. Durch <SPACE> können Sie weiterlisten lassen und durch <RUN/STOP> das Listen abbrechen.

Wenn Sie einen Drucker besitzen, können Sie das Programm auch mit <CTRL P> ausdrucken. Mit <CTRL L> wird das Programm noch einmal neu in Ihren C 64 geladen.

(F. Lonczewski/N. Mann/D. Weineck/ef)

SONDERHEFT 46

VORSCHAU



Praktische Anwendungen aus verschiedenen Bereichen stehen im Mittelpunkt des Sonderhefts 46.

Mit dem »Auto-Haushaltsbuch« behalten Sie den Überblick über alle Ausgaben fürs Auto. Künstliche Intelligenz mit dem C64: Der Computer im Dialog mit dem Anwender. Unser »Expertensystem« macht's möglich.

Üben und Lernen wird zum Vergnügen: Mit »Ede 64« lernen Kinder das Einmaleins in spielerischer Form. Chemische Grundlagen vermittelt der »Chemiekasten«.

Und der Knüller: Die Diskette mit allen Anwendungsprogrammen befindet sich im Heft. Die mühsame Arbeit des Abtippens ist nicht mehr nötig.

Das Sonderheft 46 liegt ab dem 29.9.1989 an Ihrem Kiosk.

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Hans-Günther Beer
Stellv. Chefredakteur: Gottfried Knechtel – verantwortlich für den redaktionellen Teil
Chef vom Dienst: Susanne Kirmaier
Redaktion: Elmar Friebe, Klaus Sonnenleiter, Andreas Greil
Redaktionsassistenz: Brigitte Bobenstetter, Helga Weber (202), Sylvia Derenthal
Hotline: Monika Welzel (640)
Mitarbeiter der Redaktion: Nikolaus Heusler, Florian Müller, Harald Beiler, Dirk Swienty

Alle Artikel sind mit dem Kennzeichen des Redakteurs (kn = Gottfried Knechtel, ef = Elmar Friebe, so = Klaus Sonnenleiter, ag = Andreas Greil) und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Art-director: Friedemann Porscha
Layout: Erich Schulze (Cheflayouter), Marian Schwarz, Johanna Schneider
Fotografie: Sabine Tennstaedt, Roland Müller
Titelgestaltung: Friedemann Porscha, Erich Schulze
Spritzgrafik: Norbert Raab, Ewald Standke
Computergrafik: Werner Nienstedt

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-415656, Telex: 862329 mut ch
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351
Österreich: Markt&Technik Ges. mbH Große Neugasse 28, A 1040-Wien, Tel. 0222/5871393, Telex: 047-132532

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleiter: Klaus Buck (180); Wolfgang Meyer (stellv.) (887)

Anzeigenleitung: Phillip Schiede (399) – verantwortlich für Anzeigen

Anzeigenformate: 1/1 Seite ist 266 Millimeter hoch und 185 Millimeter breit (2 Spalten à 86 Millimeter oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter.

Anzeigenpreise: Es gilt die Anzeigenpreisliste vom 5. Januar 1988. 1/1-Seite sw: DM 5400,-. Farbzuschlag: erste und zweite Farbe aus der Europa-Skala je DM 1000,-. Vierfarbzuschlag DM 2800,-. Plazierung innerhalb der redaktionellen Beiträge. Mindestgröße 1/4-Seite.

Anzeigenverwaltung und Disposition: Monika Burseg (147)

Anzeigen-Auslandsvertretung: England: F. A. Smyth&Associates Limited, 23a, Aylmer Parade, London, N2 OPQ. Telefon:0044/1/3405058, Telefax: 0044/1/3419602
Taiwan: Third Wave Publishing Corp., 1-4 Fl. 977 Min Shen E. Road, Taipei 10581, Taiwan, R.O.C., Tel. 00886/2/7630052, Telefax: 00886/2/7658767, Telex: 078529335

Vertriebsleiter: Helmut Grünfeldt (189)

Verkaufsleiter Abonnement: Benno Gaab (740)

Verkaufsleiter Einzelhandel: Robert Riesinger (364)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 14,–

Druck: SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-776.

Redaktion Sonderhefte

Redaktionsdirektor: Michael M. Pauly

Vorstand: Otmar Weber (Vors.), Bernd Balzer, Richard Kerler

Leiter Unternehmensbereich »Populäre Computerzeitschriften«: Eduard Heilmayr, Werner Pest

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0, Telex 5-22052

ISSN 0931–8933

Telefon-Durchwahl im Verlag: Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

Mitteilung gemäß Bayerischem Pressegesetz:
Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Carl-Franz von Quadt (Vorsitzender), Dr. Robert Dissmann (stellv. Vorsitzender), Ursula Berndt




GEOS-
APPLIKATIONEN

